

**RANCANG BANGUN APLIKASI**  
***MOBILE-RESTO RESERVATION PADA MULTIPLATFORM***

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Pada  
Jurusan Teknik Informatika

Oleh :

**Riswan Ahmad**  
**10851001461**



**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU**  
**PEKANBARU**  
**2013**

# ***DESIGN MOBILE- RESTO APPLICATION RESERVATION ON MULTI-PLATFORM***

**RISWAN AHMAD**

**10851001461**

*Final Exam Date: July 4<sup>th</sup> 2013*

*Graduation Ceremony Period: November 2013*

*Information Engineering Department*

*Faculty of Sciences and Technology*

*State Islamic University of Sultan Syarif Kasim Riau*

## ***ABSTRACT***

*The new technology will facilitate human in any case, among the emerging new technology is the smartphone. Smartphones can be taken and used in the search for a restaurant. Some restaurants have a less than satisfactory service in terms of ordering food and drinks. Customer complaints will be not so good service, not friendly, too late and so forth can be overcome by using smartphone technology as a medium of booking. Ordering a meal and drinks with mobile smartphones will relieve customer dissatisfaction. Restaurants are no longer using the manual method, but by utilizing the latest technology to support the business and add a sense of special interest for the customer life customer electronic (e-life). Authors make an application that can make your reservation by using the smartphone food. JQueryMobile technology helped in the completion of the design of mobile applications-resto reservation. Client and server-based applications. Use as a wireless data transfer media. Mobile-restaurant reservation application running on two platforms Android and Blackberry. All functions are running properly designed. Only a few features that do not work, ie the Blackberry platform. Due to the limitations of the device in the process of data processing. In the web browser application runs perfectly without any problems.*

**Key words:** *Business, Mobile, Reservation, Smartphone.*

# **RANCANG BANGUN APLIKASI**

## ***MOBILE-RESTO RESERVATION PADA MULTIPLATFORM***

**RISWAN AHMAD**

**10851001461**

Tanggal Sidang: 4 Juli 2013

Periode Wisuda: Nopember 2013

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

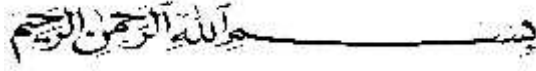
Universitas Islam Negeri Sultan Syarif Kasim Riau

### **ABSTRAK**

Teknologi baru akan mempermudah manusia dalam hal apapun, diantara teknologi baru yang muncul ialah adanya *smartphone*. Smartphone dapat dibawa dan digunakan dalam pencarian restoran. Beberapa restoran memiliki pelayanan yang kurang memuaskan dalam hal pemesanan makanan dan minuman. Keluhan pelanggan akan pelayanan yang tidak begitu baik, tidak ramah, terlambat dan lain sebagainya dapat diatasi dengan memanfaatkan teknologi *smartphone* sebagai media pemesanan. Pemesanan makan dan minuman dengan *mobile smartphone* akan mengurangi rasa ketidak puasan pelanggan. Restoran tidak lagi menggunakan cara *manual*, tetapi dengan memanfaatkan teknologi demi menunjang bisnis dan menambah rasa ketertarikan pelanggan khusus bagi pelanggan yang *electronic life* (e-life). Penulis membuat sebuah aplikasi yang dapat melakukan pemesanan makanan dengan menggunakan *smartphone*. Teknologi JQueryMobile membantu dalam penyelesaian rancang bangun aplikasi *mobile-resto reservation*. Aplikasi berbasis *client* dan *server*. Menggunakan *wireless* sebagai media tranfer data. Aplikasi *mobile-resto reservation* berjalan pada dua *platform* Android dan Blackberry. Seluruh fungsi yang dirancang berjalan dengan semestinya. Hanya sedikit fitur yang tidak berkerja, yaitu pada *platform* Blackberry. Dikarenakan keterbatasan *device* dalam melakukan proses olah data. Pada *web browser* aplikasi berjalan dengan sempurna tanpa ada permasalahan.

**Kata kunci:** *Bisnis, Mobile, Reservation, Smartphone.*

## KATA PENGANTAR



*Assalamu'alaikum Wr. Wb*

*Alhamdulillah Rabbil Alamin* penulis ucapkan sebagai tanda syukur yang dalam kepada Allah SWT. Tuhan sekalian alam, yang telah memberikan banyak nikmat, rahmat serta hidayahnya, sehingga penulis dapat melaksanakan penelitian dan menyelesaikan Tugas Akhir ini.

Shalawat serta salam terucap buat junjungan alam yaitu Nabi Muhammad SAW, karena jasa Beliau yang telah membawa manusia dari zaman kebodohan ke zaman yang penuh ilmu pengetahuan seperti sekarang ini.

Laporan Tugas Akhir yang berjudul “ **Rancang Bangun Aplikasi Mobile-Resto Reservation Pada Multiflatform**” . Laporan tugas akhir ini merupakan salah satu prasyarat untuk memenuhi persyaratan akademis dalam rangka meraih gelar kesarjanaan di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau (UIN SUSKA Riau). Judul tersebut terinspirasi pada saat melakukan kuliah pada Universitas Islam Negeri Sultan Syarif Kasim Riau. Serta dari beberapa pihak yang membantu penulis, rekan sesama kuliah Aritha Handrico, ST. yang menambahkan hal hal terkait mengenai penelitian Tugas Akhir ini.

Jika dilihat dari isinya, laporan Tugas Akhir ini terlihat sederhana, namun banyak pihak – pihak yang terlibat dibalik penyelesaian penulisan karya ini. Bagi penulis, sejak proses perkuliahan hingga sampai kepada penyelesaian laporan Tugas Akhir ini, lebih merupakan pengalaman yang sangat berharga. Dimana didalamnya terdapat sarat dengan kesan dan penuh liku-liku kehidupan yang dapat dijadikan pelajaran serta hikmah yang sangat bermakna dan bernilai bagi penulis sendiri.

Penghargaan dan rasa terimakasih yang penuh ikhlas, penulis sampaikan kepada bapak Nazruddin Safaat, MT. yang berperan sebagai pembimbing dalam penyelesaian laporan Tugas akhir ini. Berkat jasa beliau yang memberikan bimbingan, saran, masukan dan bahkan motivasi ketika penulis melaksanakan pembuatan laporan Tugas Akhir ini. Jika masih terdapat kekurangan dalam penulisan laporan Tugas Akhir ini, itu karena lebih disebabkan kelemahan penulis sendiri dalam mencerna dan menyerap apa yang telah disampaikan oleh pembimbing.

Penulis mengharapkan hasil penelitian ini dapat bermanfaat bagi semua pihak yang menunjang untuk kemajuan teknologi terutama di Negara kita Indonesia.

Penghargaan dan rasa terimakasih yang tulus juga penulis sampaikan kepada :

1. Bapak Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibu Dra. Hj. Yenita Morena, M.si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibu Dr. Okfalisa, ST, MSc selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN SUSKA RIAU.
4. Bapak Benny Sukma Negara, MT selaku dosen penguji 1 yang banyak membantu dan memberi masukan penulis dalam penyempurnaan laporan Tugas Akhir ini.
5. Bapak Iwan Iskandar, MT selaku dosen Penguji 2. Terima kasih banyak atas dukungan-dukungan, nasehat, kebaikan-kebaikan, kesabaran dan semangat yang bapak berikan kepada penulis.
6. Bapak Reski Mai Candra, ST. M.Sc sebagai koordinator tugas akhir yang telah banyak membantu dalam menyusun jadwal dan koordinasi dengan para pembimbing.

7. Seluruh staff dosen dan karyawan Fakultas Sains dan Teknologi, khususnya Jurusan Teknik Informatika UIN SUSKA Pekanbaru.
8. Terimakasih untuk kekasihku yang selalu mengingatkan akan waktu dan membantu dalam tugas akhir ini.
9. Kepada keluargaku di TIF-D 08 Aritha, Ajo leboi, keiru dreej, beng beng, boyke, Eday, alfi, ika, jannah, inar, hera, meri, febria, ami, rio, udin, tomas, maz eko, jajaj, rizki, Mr. tod, erzi, ucap jkt48, aad, kido shikiso, hendri totuit, geo, tomy cuy, ari, mas imam dan dedi. Terimakasih atas bantuan, dorongan dan motivasi yang penuh untuk menyelesaikan laporan tugas akhir ini.
10. Teman-teman seperjuangan di Jurusan Teknik Informatika khususnya angkatan 2008, terimakasih atas saran, kritik dan diskusi yang sangat membangun,
11. Semua Anggota Kiddy.inc yang tidak dapat Penulis sebutkan satu persatu.
12. Semua pihak yang terlibat baik langsung maupun tidak langsung dalam pelaksanaan tugas akhir ini yang tidak dapat penulis sebutkan satu persatu.

Akhirnya dengan penuh kesadaran diri dan dengan segala kerendahan hati, penulis menyadri bahwa hanya Allah SWT, lah yang memiliki segala kesempurnaan. Oleh karena itu penulis senantiasa mengharapkan kritik dan saran membangun dari semua pihak dan pembaca laporan Tugas Akhir ini. Kritik dan saran tersebut dapat Pembaca sampaikan ke alamat e-mail penulis: iwan.corp@gmail.com. Akhir kata penulis berharap semoga laporan ini bermanfaat bagi semua pihak, khususnya dibidang pendidikan Teknik Informatika.

Pekanbaru, Juni 2013

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL LAPORAN .....</b>	<b>i</b>
<b>LEMBAR PERSETUJUAN .....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>iii</b>
<b>LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....</b>	<b>iv</b>
<b>LEMBAR PERNYATAAN .....</b>	<b>v</b>
<b>LEMBAR PERSEMBAHAN .....</b>	<b>vi</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>ABSTRAK .....</b>	<b>viii</b>
<b>KATA PENGANTAR.....</b>	<b>ix</b>
<b>DAFTAR ISI .....</b>	<b>xii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xv</b>
<b>DAFTAR TABEL .....</b>	<b>xviii</b>
<b>DAFTAR ISTILAH .....</b>	<b>xxii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>I-1</b>
1.1. Latar Belakang .....	I-1
1.2. Batasan Masalah.....	I-4
1.3. Rumusan Masalah .....	I-4
1.4. Tujuan Penelitian .....	I-5
1.5. Sistematika Penulisan .....	I-5
<b>BAB II LANDASAN TEORI .....</b>	<b>II-1</b>
2.1. Reservasi .....	II-1
2.2. <i>Wireless Fidelity</i> (WI-FI) .....	II-2
2.3. <i>Smartphone</i> .....	II-3
2.3.1. Android.....	II-4
2.3.1.1. Android SDK ( <i>Software Development Kit</i> ) .....	II-5
2.3.2. Blackberry .....	II-6

2.3.2.1. Blackbery SDK ( <i>Software Development Kit</i> ) .....	II-8
2.4. <i>HyperText Markup Language 5 (HTML5)</i> .....	II-8
2.5. <i>Cascading Style Sheet 3 (CSS3)</i> .....	II-10
2.6. <i>JqueryMobile</i> .....	II-12
2.7. CodeIgniter .....	II-13
2.7.1. Fitur CodeIgniter .....	II-14
2.8. Model Pengembangan Perangkat Lunak <i>Sekuensial Linier</i> .....	II-15
2.9. <i>Unified Modeling Language (UML)</i> .....	II-17
1. <i>Diagram Use Case</i> .....	II-18
2. <i>Diagram Class</i> .....	II-19
3. <i>Diagram Sequence</i> .....	II-20
4. <i>Diagram Activity</i> .....	II-21
5. <i>Diagram Component dan Diagram Deployment</i> .....	II-22
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>III-1</b>
3.1. Metode Pendekatan dan Pengembangan Sistem .....	III-1
3.1.1. Metode Pendekatan Sistem .....	III-1
3.1.2 Metode Pengembangan Sistem .....	III-1
3.1.2.1 Pengumpulan Materi .....	III-2
3.1.2.2 Analisa dan Perancangan .....	III-3
3.1.2.3 Implementasi .....	III-4
3.1.2.4 Pengujian Sistem .....	III-4
3.1.3 Alat Bantu Analisis dan Perancangan .....	III-5
<b>BAB IV ANALISA DAN PERANCANGAN .....</b>	<b>IV-1</b>
4.1. Gambaran Umum Sistem .....	IV-1
4.1.1. Proses Bisnis <i>Mobile-Resto Reservation</i> .....	IV-3
4.1.2. <i>Passkey</i> .....	IV-5
4.2. Rancangan Sistem .....	IV-6
4.2.1. Perancangan Model Sistem .....	IV-7
4.2.1.1. <i>Use Case Diagram</i> .....	IV-7



4.2.1.2 <i>Usecase Diagram Spesification</i> .....	IV-8
4.2.1.3. <i>Class Diagram</i> .....	IV-15
4.2.1.4. <i>Activity Diagram</i> .....	IV-17
4.2.1.5 <i>Sequence Diagram</i> .....	IV-18
4.2.1.6 <i>Deployment Diagram</i> .....	IV-19
4.2.2. Perancangan Struktur Menu .....	IV-19
4.2.3. Perancangan Antarmuka Sistem .....	IV-23
4.2.3.1. Tampilan <i>Login</i> Dekstop .....	IV-23
4.2.3.2. Tampilan <i>Content</i> Dekstop .....	IV-24
4.2.3.3. Tampilan <i>Login Client</i> .....	IV-25
4.2.3.4. Tampilan <i>Content Client</i> .....	IV-26
<b>BAB V IMPLEMENTASI DAN PENGUJIAN</b> .....	<b>V-1</b>
5.1. Batasan Implementasi .....	V-1
5.2. Lingkungan Operasional .....	V-2
5.3. Lingkungan Implementasi .....	V-3
5.3.1. Instalasi Aplikasi Pada <i>Smartphone</i> .....	V-4
5.3.2. <i>Interface</i> Pada Sisi <i>Server</i> .....	V-6
5.4. Tahapan Pengujian .....	V-7
5.4.1. Pengujian Blackbox <i>Mobile-Resto Reservation</i> .....	V-8
<b>BAB VI PENUTUP</b> .....	<b>VI-1</b>
6.1. Kesimpulan .....	VI-1
6.2. Saran .....	VI-2
<b>DAFTAR PUSTAKA</b>	
<b>DAFTAR LAMPIRAN</b>	
<b>DAFTAR RIWAYAT HIDUP</b>	

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Revolusi informasi telah kita rasakan dengan adanya teknologi yang canggih. Penggabungan antara teknologi komputer dan telekomunikasi telah menghasilkan suatu revolusi dibidang sistem informasi (simarmata, 2006). Seiring dengan kemajuan teknologi tersebut membentuk era baru pada dunia globalisasi informasi. Di era globalisasi informasi, muncul sebuah perangkat komunikasi yang cerdas yaitu *Smartphone*. Pengguna *smartphone* sangat banyak di Indonesia, dan ini mejadi gaya hidup baru. Gaya hidup *electronic life* atau disingkat *e-life*, sudah menjadi kebiasaan masyarakat di Indonesia. Sehingga banyak hal yang mempengaruhi dalam kegiatan dan aktivitas dengan gaya *e-life* tersebut. Terutama dalam mencari kuliner pada suatu *cafe*, rumah makan, warung jajanan yang memiliki akses *wifi*.

Para pengguna *smartphone* memiliki banyak aplikasi yang membantu untuk aktivitas mereka. Misal jika mereka ingin mencari makanan, sudah terdapat aplikasi yang membantu para *smartphone* untuk mencari lokasi restoran terdekat. Sebuah restoran masih menggunakan cara *manual* untuk memesan *menu* makanan dan minuman. Cara manual ini memiliki kelemahan-kelemahan seperti sedikitnya pelayan restoran, sehingga membuat pelanggan menunggu, menggunakan banyak kertas untuk pemesanan, jika pelayan sibuk maka pelanggan harus menunggu dengan waktu yang relatif lama dan lainnya. Sangat meresahkan bagi para pelanggan yang benar-benar mengutamakan arti komitmen waktu (Pamudji, 2009).

Maka dari itu, dengan memanfaatkan gaya hidup yang *e-life* dan adanya kemajuan teknologi penulis membuat suatu aplikasi yang membantu untuk mengatasi permasalahan tersebut. Memanfaatkan teknologi *smartphone* penulis merancang aplikasi *mobile-menu reservation* sebagai media untuk pemesanan

*menu* makanan pada restoran. Aplikasi ini membantu mengurangi permasalahan diatas, dan merupakan teknologi baru dalam hal memesan sebuah *menu* makanan dan minuman pada restoran. Banyak aplikasi *mobile* telah memasuki area bisnis karena kebutuhan dan kecepatan, ketepatan, keakuratan semua data baik keuangan dan lainnya bagi perusahaan dan kostumer, sehingga aplikasi *mobile* sangat berguna bagi perkembangan bisnis. Para *developer* banyak yang membangun aplikasi penunjang bisnis yang sangat menjanjikan. Bisnis dan teknologi informasi yang terintergrasi akan membuat perusahaan semakin berkualitas dalam setiap pengelolaan karyawan maupun manajemen perusahaan.

Aplikasi *mobile-resto reservation* berbasis *client* dan *server*. Untuk aplikasi klien berjalan di *multiflatform*, sehingga bisa implementasikan pada banyak model *smartphone*. Aplikasi *client* pada *smartphone* pelanggan, dan terhubung pada sebuah sistem yang ada pada *server*. Dengan menggunakan teknologi HTML5 maka sebuah sistem *menu reservation* kuliner ini berjalan diberbagai *platform*. Berbagai perangkat *mobile* telah banyak menggunakan HTML5 seperti, pengembangan aplikasi facebook telah menggunakan HTML5. Pada HTML5 memiliki fitur baru, seperti unsur *canvas* untuk gambar, berbagai *control form*, elemen konten yang lebih spesifik, dukungan terhadap penyimpanan *offline*, serta dukungan untuk *multimedia*. Selain fitur banyak elemen – elemen baru yang terdapat pada HTML5 yang menunjang untuk pembuatan sistem ini. Maka dari itu penulis memilih HTML5 sebagai bahasa pemrograman pendukung dalam pembuatan sistem ini. Pada proses koneksi, maka yang digunakan adalah *wifi*. Sehingga pelanggan yang datang tinggal mengkoneksikan *smartphone* mereka pada *wifi* yang disediakan oleh pihak restoran. Setelah terhubung pelanggan tinggal melakukan *order* pemesanan *menu* yang ingin dipesan.

Penulis memilih *multiplatform client* untuk pemesanan *menu* makanan, karena di Indonesia sudah sangat banyak *smartphone* yang digunakan oleh masyarakat. Seperti Blackberry, Android, Symbian dan Ios adalah sistem operasi *mobile* yang banyak diminati oleh masyarakat Indonesia. Menurut website *idc.com*, *International Data Corporation* sebuah *website* yang menangani akan

*survey* data pengguna *smartphone* di dunia menunjukkan bahwa pengguna *smartphone* yang paling terbesar adalah pada Android yaitu 75% dunia, selanjutnya pada posisi kedua adalah Ios 14.9% dan ketiga adalah Blackberry yaitu 4.3% (*source: IDC Worldwide Mobile Phone Tracker*, November 1, 2012).

Penelitian yang sudah ada dan terkait dengan aplikasi *mobile-resto* ini adalah;

- a. Penelitian Mahendra, Filo Adi Cahya Mengenai sistem “Pemesanan Makanan Otomatis Pada Rumah Makan Berbasis Mikrokontroler AT89S51”. Penelitian ini adalah untuk merancang dan membuat sistem pemesanan makanan pada rumah makan dengan menggunakan modul sistem minimum AT89S51.
- b. Penelitian Angelina, Jacky, dan Teddy Gunawan “Aplikasi Portal Pemesanan Makanan Pada Smartphone Android”. Penelitian ini adalah untuk merancang suatu aplikasi *mobile* yang mampu memberikan informasi *menu* makanan dari berbagai restoran secara *detail* dan *up to date* kepada pengguna.
- c. Penelitian Reni Sito Wulandari “Analisis Dan Perancangan Sistem *Mobile Delivery Order* Berbasis Android Di Shania *Cake Shop* Yogyakarta”. Penelitian ini membuat aplikasi pemesanan makanan secara *online* pada sebuah *cake shop*.

Penulis mencantumkan beberapa skripsi terkait mengenai penelitian yang akan dibuat. Skripsi Mahendra, Filo Adi Cahya adalah pembuatan pemesanan menu makanan menggunakan IC *AT89S51*. Mikrokontroler-lah yang mengatur akan proses sebuah pemesanan yang ada pada sistem. Skripsi Angelina, Jacky, dan Teddy Gunawan (2012), adalah sebuah portal pemesanan makanan diberbagai macam restoran dan bekerja secara online. Sedangkan skripsi Reni Sito Wulandari (2012) aplikasi pemesanan makanan pada sebuah *cake shop* yang ada di Yogyakarta menggunakan *native java platform android*. Perbedaan penelitian diatas dengan Penelitian yang akan penulis buat adalah, *menu-resto reservation* ini tidaklah *online*, melainkan terkoneksi secara *wifi*, serta aplikasi *client* dibuat

berjalan di *multiplatform*. Dilihat skripsi terkait diatas sistem berjalan secara *online* dan menggunakan jaringan *internet*. Serta aplikasi *client* dibangun dengan bahasa pemrograman *java* yang dikhususkan pada android. Penelitian baru yang akan penulis buat yaitu dengan berbagai *platform* dan bukan *android* saja, melainkan sistem operasi *mobile* lain yang banyak digunakan oleh masyarakat Indonesia. Pembuatan sistem dengan teknologi HTML5, sehingga bagaimana merancang aplikasi dengan menggunakan teknologi HTML5 diberbagai *platform mobile*.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan diatas, maka dapat diambil suatu rumusan masalah yaitu: “Bagaimana merancang bangun aplikasi *mobile-resto reservation* kuliner pada *multiplatform* berbasis *client server*.”

## **1.3. Batasan masalah**

Berdasarkan latar belakang dan rumusan masalah diatas, maka penulis membuat batasan permasalahan sebagai berikut :

1. Aplikasi hanya untuk pemesanan *menu* kuliner *base client* dan *server*.
2. Koneksi yang digunakan antara *client* dan *server* adalah *wifi*.
3. Pemesanan makanan yang dimaksud bukanlah *online* tetapi pelanggan berada pada restoran.
4. Aplikasi dibangun tanpa adanya keamanan yang lebih jika terjadi *spam* atau *hack* dari pihak lain.
5. Aplikasi *mobile menu reservation* menggunakan *passkey*

#### **1.4. Tujuan**

Tujuan dan manfaat dari pembuatan tugas akhir ini adalah dihasilkannya aplikasi *mobile-resto reservation* kuliner pada suatu restoran, sehingga lebih praktis dan efisien dalam proses pemesanan kuliner. Serta dihasilkannya suatu sistem aplikasi baru untuk menunjang proses bisnis pada sebuah restoran, demi kelancaran dan kenyamanan konsumen.

#### **1.5. Sistematika Penulisan**

Sistematika penulisan tugas akhir ini dibagi menjadi 6 (enam) bab. Berikut penjelasan tentang masing-masing bab :

#### **BAB I PENDAHULUAN**

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

#### **BAB II LANDASAN TEORI**

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian, yang terdiri dari pengertian aplikasi, alat bantu perancangan sistem aplikasi, Basis data, dan rancangan dasar berbasis *client server*.

#### **BAB III METODOLOGI PENELITIAN**

Berisikan tentang urutan atau tahapan pelaksanaan penelitian dimulai dari awal hingga laporan ini selesai.

## BAB IV ANALISIS DAN PERANCANGAN

Bab ini membahas tentang analisa sistem, karakteristik pengguna sistem, alat bantu analisis dan *modeling*, *webserver*, basis data, dan perancangan antar muka. Serta komunikasi antar perangkat yaitu pada *client* dan *server*.

## BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas mengenai implementasi *Mobile-menu* kuliner pada *platform* Android dan Blackberry, *base client server*, implementasi perangkat lunak, lingkungan implementasi, pengujian sistem, dan hasil pengujian.

## BAB VI PENUTUP

Bab ini merupakan bab terakhir yang menguraikan tentang kesimpulan yang diperoleh, serta saran bagi pengembangan program ini.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Reservasi**

Pemesanan dalam bahasa Inggris adalah *Reservation* yang berasal dari kata “*to reserve*” yaitu menyediakan atau mempersiapkan tempat sebelumnya. Sedangkan *reservation* yaitu pemesanan suatu tempat fasilitas.

Pengertian reservasi adalah sebuah proses perjanjian berupa pemesanan sebuah produk baik barang maupun jasa dimana pada saat itu telah terdapat kesepahaman antara konsumen dengan produsen mengenai produk tersebut, namun belum ditutup oleh sebuah transaksi jual beli. Pada saat reservasi berlangsung biasanya ditandai dengan adanya proses tukar-menukar informasi antara konsumen dan produsen agar kesepahaman mengenai produk dapat terwujud (Suartana, 1987).

Alasan reservasi menjadi sebuah media yang sangat efektif baik bagi produsen maupun bagi konsumen adalah produsen akan dapat melakukan evaluasi terhadap produk yang akan mereka jual melalui tingkat tinggi rendahnya jumlah reservasi jauh sebelum produk tersebut dijual (barang) ataupun diselenggarakan (jasa), dimana hasil evaluasi tersebut akan membantu produsen untuk menentukan langkah pemasaran yang akan diambil terhadap produk yang akan dijual tersebut (Suartana, 1987).

Sedangkan bagi konsumen melalui media reservasi dapat menimbang terlebih dahulu sebelum membeli produk sampai dengan waktu yang telah ditentukan (*time limit*). Sampai dengan *time limit* yang telah ditentukan produk yang telah dipesan tersebut tidak boleh dijual kepada konsumen lain, karena secara tertulis maupun tidak sampai dengan *time limit* produk tersebut telah diprioritaskan kepada konsumen yang telah melakukan reservasi. Hal ini akan sangat menguntungkan konsumen karena sampai dengan *time limit* yang



ditentukan, konsumen dapat melakukan perbandingan dengan produk lain. Selain itu, jika terjadi sesuatu dan lain hal yang mengharuskan konsumen untuk batal membeli produk tersebut, konsumen yang bersangkutan tidak harus membeli produk tersebut (Suartana, 1987).

Layaknya perjanjian lainnya, reservasi akan dinyatakan batal apabila terjadi beberapa hal berikut (1) Terjadi sesuatu di luar kendali manusia (bencana alam, perang, dll) sehingga produk tidak dapat dibuat atau diselenggarakan oleh produsen, (2) Konsumen melakukan pembatalan sebelum batas waktu, (3) Terdapat kesalahan pengisian data penting mengenai konsumen pada saat proses reservasi sehingga produsen harus membatalkan reservasi (catatan: tergantung kebijakan masing-masing perusahaan), (4) Tidak dilakukan transaksi terhadap produk yang dipesan sampai dengan *time limit* yang telah ditentukan, (5) Baik produsen maupun konsumen melakukan hal-hal yang menyebabkan reservasi harus dibatalkan, tergantung perjanjian yang dibuat atau ditentukan (Suartana, 1987).

## **2.2. Wireless fidelity (Wi-Fi)**

*Wireless Fidelity* (Wi-Fi) adalah koneksi tanpa kabel seperti *handphone* dengan mempergunakan teknologi radio sehingga pemakainya dapat mentransfer data dengan cepat dan aman. Wi-Fi tidak hanya dapat digunakan untuk mengakses internet, Wi-Fi juga dapat digunakan untuk membuat jaringan tanpa kabel di perusahaan. Karena itu banyak orang mengasosiasikan Wi-Fi dengan “Kebebasan” karena teknologi Wi-Fi memberikan kebebasan kepada pemakainya untuk mengakses internet atau mentransfer data dari ruang *meeting*, kamar hotel, kampus, dan *cafe-cafe* yang bertanda “Wi-Fi Hot Spot”. Juga salah satu kelebihan dari Wi-Fi adalah kepraktisan, tidak perlu memasang kabel *network*.

Wi-Fi merupakan kependekan dari *Wireless Fidelity*, memiliki pengertian yaitu sekumpulan standar yang digunakan untuk jaringan lokal nirkabel (*Wireless Local Area Networks* - WLAN) yang didasari pada spesifikasi *Institute of*

*Electrical and Electronics Engineers* (IEEE 802.11). Standar terbaru dari spesifikasi 802.11a atau b, seperti 802.16 g, saat ini sedang dalam penyusunan, spesifikasi terbaru tersebut menawarkan banyak peningkatan mulai dari luas cakupan yang lebih jauh hingga kecepatan transfernya (Zakaria, 2008).

Awalnya Wi-Fi ditujukan untuk penggunaan perangkat nirkabel dan *Local Area Network* (LAN), namun saat ini lebih banyak digunakan untuk mengakses internet. Hal ini memungkinkan seseorang dengan komputer dengan kartu nirkabel (*wireless card*) atau *Personal Digital Assistant* (PDA) untuk terhubung dengan internet dengan menggunakan titik akses (atau dikenal dengan hotspot) terdekat (Zakaria, 2008).

Ada 2 *mode* operasi Wi-fi, yang pertama *Mode Managed* dengan menggunakan *Access Point* yang berfungsi mengatur lalu lintas data, *access point* ini berfungsi seperti pusat *transfer* data. Kelebihan mode ini adalah *transfer* data lebih cepat, *area coverage* yang lebih luas, pengaturan dan keamanan data lebih terjamin. *Mode* ini biasa digunakan untuk hotspot wifi dan perkantoran yang memerlukan stabilitas dan keamanan data (Zakaria, 2008). Yang kedua adalah *Mode Ad-Hoc* yaitu koneksi antar *device Wi-fi peer too peer*, kelemahan mode ini adalah bila *device* wifi yang terhubung terlalu banyak, *transfer* data jadi lambat. Keuntungannya, lebih murah dan praktis bila yang terkoneksi cuma 2 atau 3, tanpa beli *access point* (Zakaria, 2008).

### **2.3. *Smartphone***

*Smartphone* adalah suatu ponsel yang memiliki kemampuan komputasi yang lebih canggih dan konektivitas melebihi kemampuan ponsel biasa. Selain itu hal mendasar yang membedakan *smartphone* dengan ponsel biasa adalah kemampuan untuk menjalankan aplikasi *third party*.

*Smartphone* memiliki processor yang mampu menjalankan beberapa fitur yang lebih aplikatif, sehingga *smartphone* yang muncul ditahun 2012 hampir

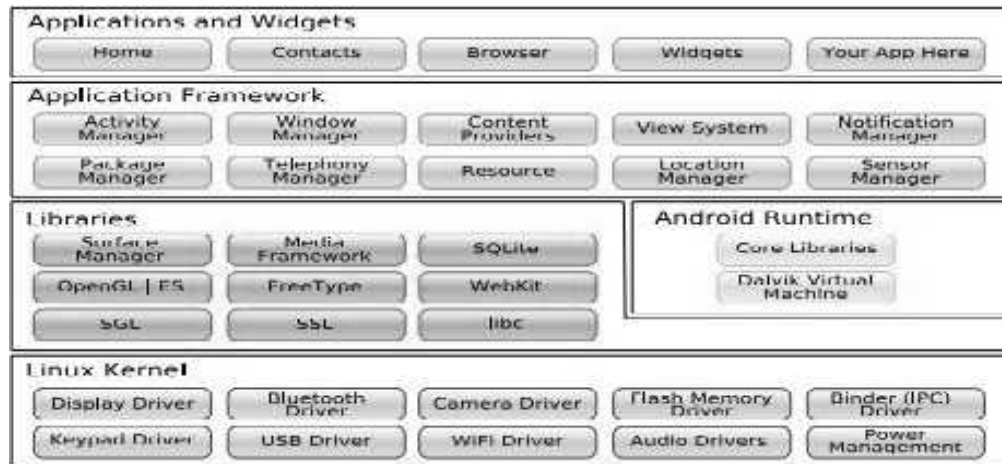
menyamai mini komputer. *Smartphone* tersebut juga memiliki *space disk*, *memory* dan sistem operasi.

### 2.3.1. Android

*Smartphone* yang banyak beredar di Indonesia menggunakan sistem operasi android. *Smartphone* dengan teknologi *touch screen* ini berkembang dengan pesat, serta perkembangan yang sangat signifikan di beberapa negara. Seperti di Indonesia pengguna android sudah mulai mencapai 30%.

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi (Safaat, 2012). Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi. Pada saat perilisan perdana android, 5 november 2007. Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*.

Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama *Cupcake*, versi 1.6 dirilis pada 15 September 2009 diberi nama *Donut*, versi 2.0/2.1 dirilis pada 26 Oktober 2009 diberi nama *Éclair*, versi 2.2 dirilis pada bulan Mei 2010 diberi nama *Froyo* dan versi 2.3 dirilis pada Desember 2010 yang diberi nama *Gingerbread*, Versi 3.0 dirilis pada Februari 2011 dengan nama *Honeycomb*. Versi 4.0 dirilis pada November 2011 dengan nama Ice Cream Sandwich. Versi 4.1 dirilis pada Juni 2012 dengan nama *Jelly Bean*. Ini adalah versi terbaru dari android.



Gambar 2.1 Arsitektur Android (Safaat, 2012)

1. *Application and Widgets* (paling atas), user hanya berinteraksi pada aplikasi seperti *download* dan *install*.
2. *Application Framework* (ke dua dari atas) adalah *layer* bagi para pembuat aplikasi.
3. *Libraries* dan *android runtime* adalah *layer* bagi aplikasi yang memiliki *database* seperti *sql-lite*.
4. *Linux Kernel* merupakan *layer* untuk *root*.

#### 2.3.1.1. Android SDK (*Software Development Kit*)

Android SDK Adalah *tool API (Application programming Interface)* yang di perlukan untuk memulai mengembangkan aplikasi pada *platform* Android. Android berjalan didalam *Dalvik Virtual Machine (DVM)* bukan di *Java Virtual Machine (JVM)*. Android SDK sebagai alat bantu dan API untuk memulai mengembangkan Aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Untuk versi *operating system* terbaru tersedia pada SDK dan dapat di-*update*. *Source* SDK Android dapat diunduh langsung pada situs resmi pengembang SDK Android.

### 2.3.2. Blackberry

BlackBerry adalah perangkat selular yang memiliki kemampuan layanan *push e-mail*, telepon, sms, menjelajah internet, *messenger* (Blackberry Messenger - BBM), dan berbagai kemampuan nirkabel lainnya. Penggunaan *gadget* canggih ini begitu fenomenal belakangan ini, sampai menjadi suatu kebutuhan untuk *fashion*. BlackBerry pertama kali diperkenalkan pada tahun 1999 oleh perusahaan Kanada, *Research In Motion* (RIM). Kemampuannya menyampaikan informasi melalui jaringan data nirkabel dari layanan perusahaan telepon genggam hingga mengejutkan dunia.

Perangkat BlackBerry pertama, 850, diperkenalkan pada tahun 1999 sebagai *pager* dua arah di Munich, Jerman. Nama BlackBerry diciptakan oleh Branding perusahaan pemasaran Lexicon. Nama itu dipilih karena kemiripan tombol *keyboard* untuk bahwa dari *drupelets* yang membentuk buah blackberry.

Pada tahun 2003, BlackBerry *smartphone* lebih dikenal dirilis, yang mendukung *push email*, telepon seluler, pesan teks, *internet faxing*, *web browsing* dan layanan informasi nirkabel. Ini adalah contoh dari perangkat konvergen. Perangkat BlackBerry asli, RIM 850 dan 857, menggunakan jaringan DataTac.

BlackBerry pertama membuat kemajuan dipasar dengan berkonsentrasi pada *email*. RIM saat ini menawarkan layanan email BlackBerry untuk non-BlackBerry perangkat, seperti Palm Treo, melalui BlackBerry-nya *Connect* perangkat lunak.

Sistem operasi yang digunakan oleh perangkat BlackBerry adalah lingkungan *multitasking proprietary* yang dikembangkan oleh RIM. Sistem operasi ini dirancang untuk penggunaan perangkat input seperti roda trek, *track ball*, dan *track pad*. OS menyediakan dukungan untuk Java MIDP 1.0 dan WAP 1.2. Versi sebelumnya memungkinkan sinkronisasi nirkabel dengan *email* Microsoft *Exchange Server* dan kalender, serta dengan *email* Lotus Domino. OS 5.0 menyediakan subset dari MIDP 2.0, dan memungkinkan aktivasi nirkabel

lengkap dan sinkronisasi dengan *Exchange*, kalender *email*, tugas, catatan, dan kontak, dan menambahkan dukungan untuk Novell GroupWise dan Lotus Notes. BlackBerry Curve 9360, BlackBerry Torch 9810, Bold 9900/9930, Curve 9310/9320 dan Torch 9850/9860 fitur BlackBerry terbaru OS 7 (diluncurkan tahun 2011).

Pengembang pihak ketiga dapat menulis perangkat lunak menggunakan API ini, dan kepemilikan API BlackBerry juga. Setiap aplikasi yang menggunakan fungsi terbatas tertentu harus ditandatangani secara digital sehingga dapat dikaitkan ke akun pengembang di RIM. Prosedur ini menjamin penandatanganan kepengarangan aplikasi tetapi tidak menjamin kualitas atau keamanan kode. RIM menyediakan alat untuk mengembangkan aplikasi dan tema untuk BlackBerry. Aplikasi dan tema dapat dimuat ke perangkat BlackBerry melalui BlackBerry App World, *Over The Air* (OTA) melalui *browser* ponsel BlackBerry, atau melalui *BlackBerry Desktop Manager*.

BlackBerry terintegrasi pada sistem *e-mail* yang terorganisasi melalui paket perangkat lunak yang disebut *BlackBerry Enterprise Sistem* (BES). BES dapat digunakan oleh jaringan *e-mail* yang berbasis Microsoft *Exchange*, Lotus Domino, dan Novell Group Wise. Khusus pada pengguna individu, mereka dapat menggunakan layanan *e-mail* nirkabel yang disediakan oleh provider tanpa harus menginstalasi BES. Perangkat lunak yang diperuntukkan bagi pengguna pribadi ini memungkinkan Anda untuk mengintegrasikan *smartphone* dengan 10 akun *e-mail* yang berbasis *Post Office Protocol* (POP3) dan *Internet Message Access Protocol* (IMAP), menerima dan mengirim pesan instan, serta berselancar di *internet*. Dengan BIS (*Blackberry Internet Service*), kita juga dapat membuka tambahan data (*attachment*) dalam bentuk excel, word, power point, pdf, zip, jpg, gif dengan tingkat kompresi data yang tinggi.

#### **2.3.2.1. Blackberry SDK (*Software Development Kit*)**

Blackberry memiliki SDK (*Software Development kit*) untuk pengembang yang ingin mengembangkan aplikasi blackberry. RIM (*Research In Motion*) juga mengeluarkan IDE (*Integrated Development Environment*) khusus yaitu JDE Blackberry. Tetapi fitur-fitur yang ada sedikit menyulitkan untuk pengembang mengembangkan aplikasi blackberry. Pengembang aplikasi blackberry dapat menggunakan IDE dari Eclipse yang lebih *user-friendly*. Memiliki fungsi yang sama yaitu membantu para pengembang aplikasi blackberry untuk membangun sebuah aplikasi. Blackberry SDK terbagi dua kategori yaitu untuk *operating system* 10 dan untuk *operating system* 7 atau dibawah 7. Jika pengembang ingin membuat aplikasi yang berjalan pada OS 10 maka gunakan SDK blackberry 10, tetapi jika pengembang ingin membuat aplikasi berjalan pada OS 7 maka gunakan SDK blackberry 7.

Blackberry memiliki WebWork yaitu sebuah *platform* pengembang untuk *web apps*. Dengan menggunakan BlackBerry *web app packager* (*WebWorks*), fitur utama BlackBerry seperti *multi-tasking*, *push technology*, *phone functionality*, *storage*, *network*, *location based* dan *service* lainnya seperti *payment service*, *advertising*, dapat diintegrasikan kedalam sebuah aplikasi hanya dengan menggunakan bahasa pemrograman yang familiar, seperti HTML5, CSS dan juga Javascript. BlackBerry WebWorks juga menyertakan beberapa *update* untuk *web application packager*, *web APIs* dan BlackBerry *Plugin* mereka. Fungsi baru dari API mereka ini menyertakan fungsi SMS, akses ke *Call Log*, dan juga kontrol *file audio*.

#### **2.4. HyperText Markup Language 5 (HTML5)**

*HyperText Markup Language* (HTML5) adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi *World Wide Web*, sebuah teknologi inti dari Internet. HTML5 adalah revisi kelima dari HTML, yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997 dan

hingga bulan Juni 2011 masih dalam pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan salah satu karya Konsortium Waring Wera Wanua (*World Wide Web Consortium*, W3C) untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat *web*.

Secara khusus, HTML5 menambahkan banyak fitur baru sintaksis. Ini termasuk `<video>` baru, `<audio>` dan elemen `<canvas>`, serta integrasi *scalable vector* grafis konten (SVG) (yang menggantikan penggunaan tag generik `<object>`) dan MathML untuk rumus matematika. Fitur-fitur ini dirancang untuk memudahkan untuk memasukkan dan menangani konten multimedia dan grafis di *web* tanpa harus resor untuk *plugin proprietary* dan API. Elemen baru lainnya, seperti `<section>`, `<article>`, `<header>` dan `<nav>`, dirancang untuk memperkaya isi semantik dokumen. Atribut baru telah diperkenalkan untuk tujuan yang sama, sementara beberapa elemen dan atribut telah dihapus. Beberapa elemen, seperti `<a>`, `<cite>` dan `<menu>` telah diubah, atau didefinisikan ulang standar. API dan dokumen model obyek (DOM) tidak *afterthoughts* lagi, tetapi bagian *fundamental* dari spesifikasi HTML5. HTML5 juga mendefinisikan secara rinci proses yang diperlukan untuk dokumen yang tidak *valid* sehingga kesalahan sintaks akan diperlakukan secara seragam oleh semua *browser conforming* dan agen pengguna lainnya.

HTML juga akan berkaitan erat dengan CSS (*Cascading Style Sheet*). CSS menentukan *format* dari konten. Dengan adanya CSS, konten *website* bisa lebih seragam sehingga hasilnya tampak cantik, dan *format* ini dapat diatur dengan cara yang berbeda (Zaki dkk, 2012). Berbicara Mengenai HTML tidak terlepas dari



javascript, yang merupakan bahasa pemrograman *client-side* untuk memberikan efek dinamis. Javascript mampu membuat *web* terlihat lebih dinamis dan interaktif. Maka dari itu HTML tidak berdiri sendiri, Sebuah halaman *web* biasanya hasil kolaborasi antara HTML, CSS dan Javascript.

HTML5 sendiri sampai saat ini masih terus dikembangkan, tapi mayoritas *browser modern* sudah bisa mendukung HTML5. Ada beberapa aturan yang diterapkan untuk HTML5 (Zaki dkk, 2012), seperti

- a. Fitur-fitur baru harus berbasis HTML, CSS, DOM, dan Javascript.
- b. Mengurangi kebutuhan *plugin* eksternal (contohnya Flash).
- c. *Error handling* yang lebih baik.
- d. *Markup* tambahan untuk menggantikan *scripting*.
- e. HTML5 harus bisa diakses dari peranti manapun.

Sintaks untuk HTML5 sangat fleksibel, namun bukan berarti anda bisa menulis sembarangan. HTML5 mendukung pemisahan yang tegas antara tahapan presentasi dan konten.

Beberapa fitur baru di HTML5 (Zaki dkk, 2012) antara lain :

- a. Elemen *canvas* untuk menggambar.
- b. Elemen *audio* dan *video* untuk memutar *audio* dan *video*.
- c. Dukungan untuk penyimpanan *local* atau *offline*
- d. Elemen yang spesifik terhadap konten, seperti *article*, *footer*, *header*, *nav* dan *section*.
- e. Kontrol *form* baru, seperti *calendar*, *date*, *time*, *email*, *url* dan *search*.

## **2.5. Cascading Style Sheet 3 (CSS3)**

*Cascading Style Sheet* (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. CSS merupakan salah satu

pemrograman wajib di samping HTML yang harus dikuasai oleh setiap pemrograman *web*, terlebih lagi itu *web designer* (Saputra, 2012).

Sama halnya *styles* dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya *heading*, subbab, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas. Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Tujuan utama dari CSS adalah untuk memisahkan konten utama dengan tampilan dokumen lainnya. Dengan adanya pemisahan ini, akses konten pada *web* meningkat. *Web* yang menggunakan CSS akan lebih ringan dan mudah untuk dibuka dibandingkan dengan *web* yang tidak menggunakan CSS. Tujuan lainnya adalah untuk mempercepat pembuatan halaman *web*. Pembuatan hanya perlu satu *property* dan *property* tersebut dapat digunakan pada elemen lainnya, dan kita tidak perlu menulis ulang kode program yang digunakan berulang kali.

CSS sudah mencapai versi 3 dimana setiap versi pasti ada peningkatan yang dilakukan. CSS-1 Masih kuno CSS hanya dikembangkan dan digunakan untuk *formatting* dokumen html. CSS-2 Mulai sedikit bagus dan mantap. Sudah menggunakan *font*, *table-layout*, dan berbagai *media* untuk *printer*. Sedangkan yang terbaru adalah CSS-3 merupakan pengembangan dari CSS sebelumnya. Peningkatan yang paling mencolok pada versi ini adalah peningkatan fitur yang mengarah pada efek animasi, seperti *crop*, *slide images*, dan lainnya.

## 2.6. JQueryMobile

JQuery *Mobile* adalah sebuah *platform* pengembangan dari jQuery. JQuery adalah *library* JavaScript yang populer untuk membangun *website* interaktif yang kaya serta bisa dipakai juga untuk aplikasi *web* (Zaki dkk, 2012). Dengan adanya perkembangan dunia *mobile*, maka jQuery pun mulai disalurkan kearah aplikasi *web mobile*.

Dengan jQuery *Mobile*, anda bisa melihat banyak hal yang sangat menarik seputar aplikasi *mobile*. Anda bisa dengan mudah menyajikan tampilan *page* yang sangat *mobile friendly* tanpa harus mengetik banyak kode.

Dengan menggunakan jQuery *mobile*, anda bisa *men-develop* berbagai solusi *mobile* yang bekerja dengan baik di berbagai peranti sistem operasi *mobile*. Contoh peranti yang didukung jQuery *Mobile* antara lain, Android, Blackberry OS6, Fennec (Mozilla), WebOS dari HP (Palm), Ios (Iphone), serta Opera *Mobile*. Platform lain seperti MeeGo, Windows *Mobile* dan platform Symbian akan didukung di masa depan.

JQuery *Mobile* adalah proyek pencabangan dari JQuery, yang mulai dipisahkan menjadi proyek independen mulai tanggal 11 Agustus 2010. Tujuannya adalah menggunakan jQuery untuk perangkat *mobile*. Developer JQuery berusaha agar JQuery bisa kompatibel dengan semua merek *browser mobile*, dimana koreksi bug dan perbaikan selalu dirilis dengan teratur. Dengan menggunakan JQuery *Mobile* kita bisa membuat aplikasi *web* yang *multiplatform*, tidak tergantung pada peranti keras tertentu. Dengan kode yang sam, aplikasi *web* kita bisa jalan di hampir semua peranti *mobile* yang populer saat ini. JQuery *Mobile* juga telah mendukung penggunaan layar sentuh, sehingga lebih optimal.

Jquery dikembangkan menggunakan sifat-sifat tertentu. Beberapa diantara sifat JQuery *Mobile* (Zaki dkk, 2012) adalah :

- a. JQuery *Mobile* memudahkan pengembangan *user-interface* untuk *mobile web apps*
- b. Konfigurasi antarmuka bersifat *markup-driven*, yang berarti anda dapat membuat seluruh aplikasi antarmuka dasar dalam kode HTML, tanpa perlu menulis satu baris JavaScript.
- c. Menyediakan serangkaian *event* khusus, mendeteksi *event* dari peranti *mobile* dan peranti *touch screen* seperti *tap*, *tap-and-hold*, *swipe* dan perubahan orientasi (misal Merotasi Peranti).
- d. Memastikan bahwa *user-interface* bekerja pada *web browser*.
- e. Terdapat theme yang memudahkan penyesuaian tampilan aplikasi.

JQuery *Mobile* juga memungkinkan untuk mengembangkan program aplikasi menjadi lebih cepat, karena aksesibilitas lebih cepat dan tidak memerlukan konfigurasi javascript. JQuery *Mobile* sangat kompatibel terhadap fitur – fitur yang disediakan.

## 2.7. CodeIgniter

CodeIgniter (CI) adalah framework pengembangan aplikasi (*Application Development Framework*) dengan menggunakan PHP. Suatu kerangka pembuatan program dengan menggunakan PHP. Pengembang dapat langsung menghasilkan program dengan cepat, dengan mengikuti kerangka kerja untuk membuat yang telah disiapkan oleh *framework* CI ini (Betha, 2012).

CI menyediakan sekumpulan librari yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses librarinya. Pemrograman dapat memfokuskan diri pada *code* yang harus dibuat untuk menyelesaikan suatu pekerjaan. CI merupakan *framework* yang memiliki dokumentasi yang jelas dan lengkap, yang memudahkan pengembang untuk mempelajari dengan mudah. CI dapat digunakan dengan menggunakan *server web apache* MS IIS. Hanya saja tidak semua dari fitur CI dapat diimplementasikan dengan sempurna di MS IIS,

karena dirancang dan dikembangkan dengan acuan *apache*. Contoh yang pasti adalah penggunaan file *.htaccess*, yang memang milik *apache*. Secara umum CI memberikan dan menyediakan (Betha, 2012) :

- a. Suatu kerangka kerja dengan jejak yang sedikit (*Small footprint*)
- b. Kinerja yang bagus
- c. Kompabilitas dengan akun *hosting* standar yang terpasang dengan perbedaan versi dan konfigurasi
- d. Suatu kerangka kerja yang hampir tidak memerlukan konfigurasi
- e. Suatu kerangka kerja yang tidak memerlukan kita menggunakan baris perintah (*command line*)
- f. Suatu kerangka kerja yang tidak mengharuskan bekerja dengan aturan pengkodean yang mengikat atau membatasi
- g. Suatu aturan yang tidak mengharuskan kita dipaksa menggunakan suatu bahasa *templating* (walaupun CI menyediakan suatu *parser* untuk *template*)
- h. Cara untuk menghindari kompleksitas dalam membuat program, dengan menyediakan solusi yang sederhana
- i. Suatu dokumentasi kerangka kerja yang menyeluruh dan jelas

#### **2.7.1. Fitur CodeIgniter**

Berikut fitur - fitur dari CI (Betha, 2012) :

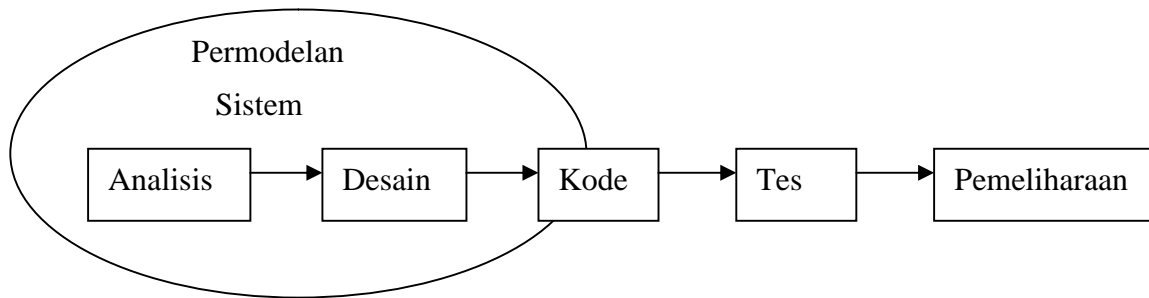
1. Sistem berbasis *model – view – controller*
2. Benar – benar *framework* yang ringan
3. Memiliki fitur *class database* yang mendukung berapa *platform*
4. Dukungan database dengan *active record*
5. *Form* dan validasi data
6. Keamanan dan *filtering XSS*
7. Manajemen sesi (*session management*)

8. *Class* untuk pengirim email yang mendukung lampiran (*attachment*), email dengan HTML/teks, menggunakan berbagai protokol (*sendmail*, SMTP, dan Mail)
9. Librari manipulasi image (menggunting, mengubah ukuran, memutar dan lainnya), mendukung GD, *imagemagick*, dan NetPBM
10. *Class* untuk *upload file*
11. *Class* FTP
12. Lokalisasi
13. Paginasi
14. Enkripsi data
15. *Benchmarking*
16. *Full page caching*
17. *Logging error*
18. *Profiling* aplikasi
19. *Class* kalender
20. *Class user agent*
21. *Class encoding* ZIP
22. *Class engine template*
23. *Class traceback*
24. Librari XML-RPC
25. *Class* unit pengujian
26. URL yang bersahabat dengan *search engine*
27. *Routing* URL yang fleksibel
28. Mendukung *Hook* dan ekstensi *class*
29. Librari fungsi “helper” yang banyak

## **2.8. Model Pengembangan Perangkat Lunak Sekuensial Linier**

Model pertama yang diterbitkan untuk proses pengembangan perangkat lunak diambil dari proses rekayasa lain (Royce, 1970). Model ini di ilustrasikan pada gambar 2.2. Berkat penurunan dari satu fase ke fase selanjutnya, model ini

dikenal sebagai ‘model air terjun’ atau siklus hidup perangkat lunak . tahap – tahap utama dari model ini memetakan kegiatan-kegiatan pengembangan dasar.



Gambar 2.2. Model Sekuensial Linear (Sommerville, 2003)

1. Analisis dan defenisi persyaratan. Pelayanan, batasan dan tujuan sistem ditentukan melalui konsultasi dengan *user* sistem. Persyaratan ini kemudian didefenisikan secara rinci dan berfungsi sebagai spesifikasi sistem.
2. Perancangan sistem dan perangkat lunak. Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan – hubunganya.
3. Implementasi dan pengujian unit. Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.
4. Integrasi dan pengujian sistem. Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirim kepada pelanggan.
5. Operasi dan pemeliharaan. Biasanya ini merupakan fase siklus hidup yang paling lama. Sistem diinstall dan dipakai. Pemeliharaan mencakup koreksi

dan berbagai *error* yang tidak ditemukan pada tahap – tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan – persyaratan baru ditambahkan.

Pada prinsipnya, hasil dari setiap fase merupakan satu atau lebih dokumen yang disetujui. Fase berikutnya tidak boleh dimulai sebelum fase sebelumnya selesai. Pada prakteknya, tahap – tahap ini bertumpang tindih dan memberi informasi satu sama lain. Pada waktu perancangan, masalah dengan persyaratan diidentifikasi, pada saat pengkodean, ditemukan masalah perancangan dan seterusnya. Proses perangkat lunak bukanlah model *linier* sederhana, tetapi melibatkan serangkaian iterasi kegiatan pengembangan.

Masalah dengan model air terjun adalah terjadinya pembagian proyek menjadi tahap – tahap yang tidak fleksibel. Komitmen harus dilakukan pada tahap awal proses, dan akan sulit bagi perekrut untuk menanggapi perubahan persyaratan pelanggan. Dengan demikian, model air terjun harus digunakan hanya ketika persyaratan dipahami dengan baik. Bagaimanapun juga, model air terjun merefleksikan praktek rekayasa. Secara konsekuen, proses perangkat lunak yang berdasarkan pada pendekatan ini masih digunakan untuk pengembangan perangkat lunak, terutama jika merupakan bagian dari sistem proyek rekayasa yang lebih besar.

## **2.9. *Unified Modeling Language (UML)***

*Unified Modeling Language* adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, 2004). Pengertian lain dari UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep



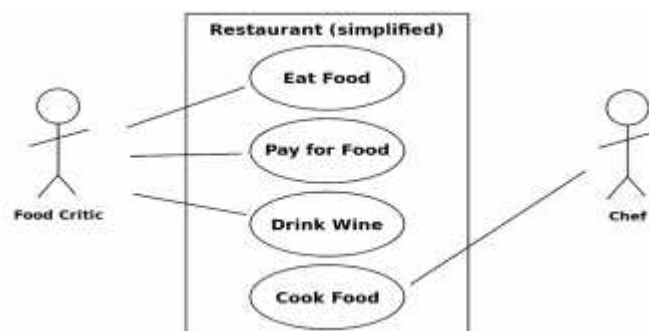
bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*.

Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh bergabung Booch untuk membuat sebuah project pendekatan metode yang *uniform* atau seragam dari masing-masing metoda mereka. Saat itu baru dikembangkan draft metoda UML version 0.8 dan diselesaikan serta di *release* pada bulan oktober 1995. UML adalah standar dunia yang dibuat oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object oriented* dan *software component*.

Alat bantu UML didefinisikan oleh beberapa *diagram*, diantaranya sebagai berikut :

#### 1. *Diagram Use Case*

*Diagram Use Case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. Menurut Whitten, (2004) *usecase diagram* adalah *diagram* yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem.



Gambar 2.3. Contoh *Usecase* (Wikipedia.org)

## 2. *Diagram Class*

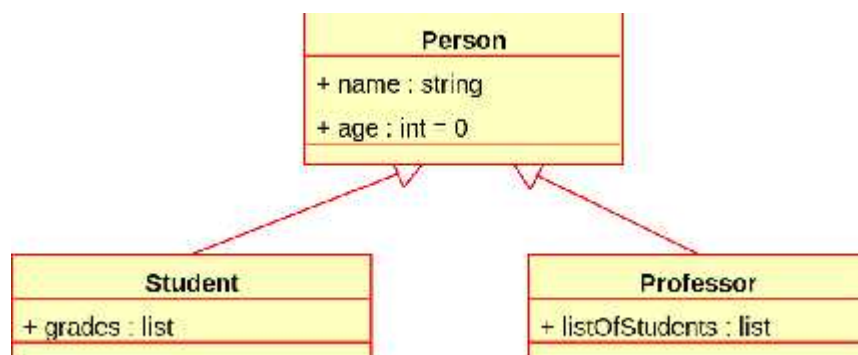
*Diagram Class* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. *Diagram Class* bersifat statis; menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan.

*Diagram Class* mempunyai 3 macam relationships (hubungan), sebagai berikut :

**Association** suatu hubungan antara bagian dari dua kelas. Terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Didalam *diagram*, sebuah *association* adalah penghubung yang menghubungkan dua kelas.

**Aggregation** suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. *Aggregation* memiliki titik pusat yang mencakup keseluruhan bagian. Sebagai contoh : *Order detail* merupakan kumpulan dari *Order*.

**Generalization** suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *superClass* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *superClass*. Contoh : *Payment* adalah *superClass* dari *Cash*, *Check*, dan *Credit*.

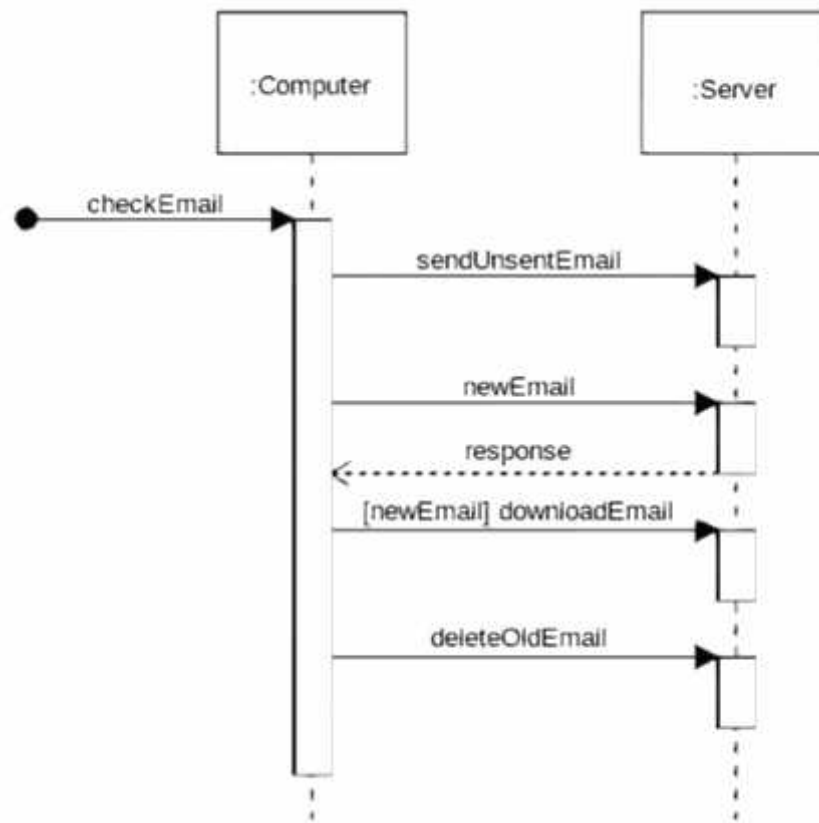


Gambar 2.4. Contoh Dari *Diagram Class* (Wikipedia.org)

### 3. *Diagram Sequence*

*Diagram Class* dan *diagram Object* merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti *Diagram Interaction*.

*Diagram sequence* merupakan salah satu *diagram Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. *Diagram* ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



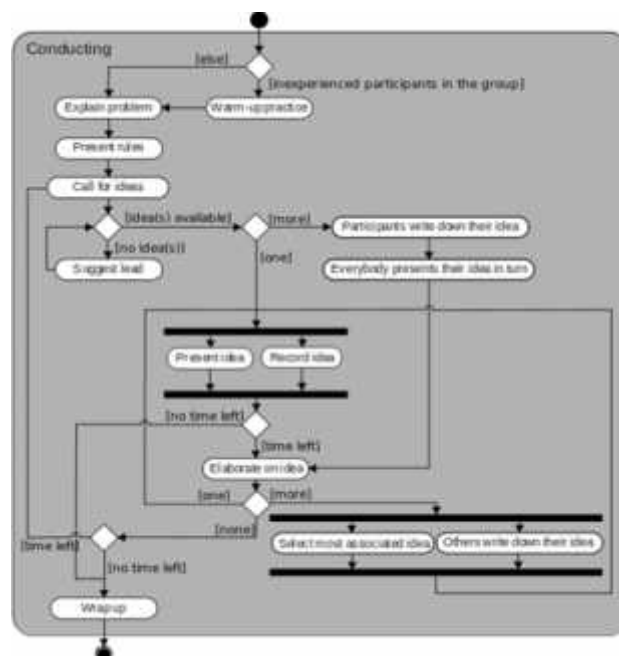
Gambar 2.5. Contoh Dari *Diagram Sequence* (Wikipedia.org)

#### 4. Diagram Activity

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

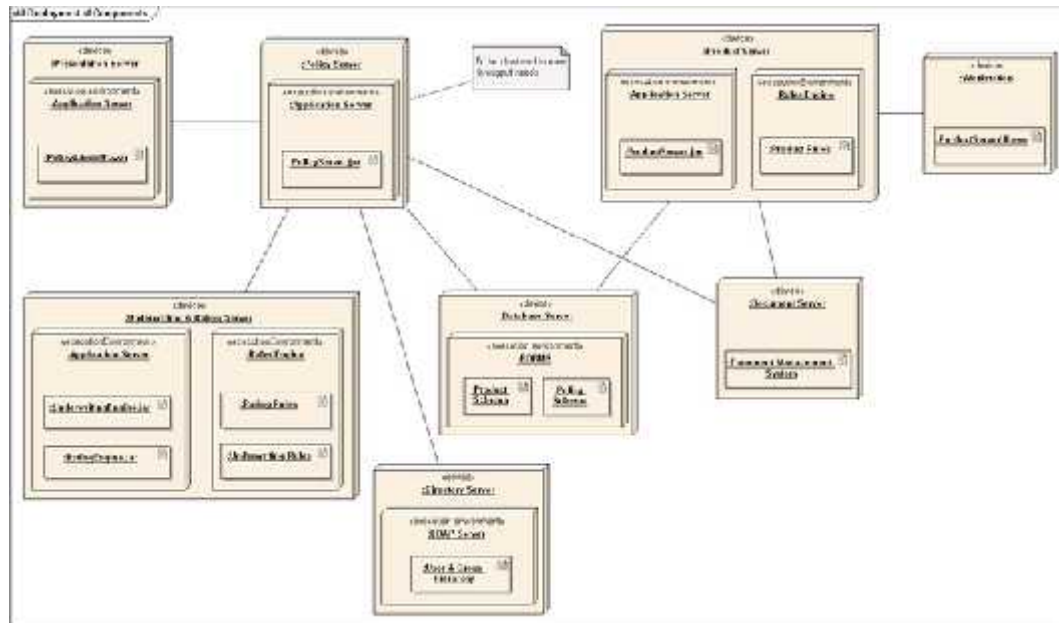
Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.



Gambar 2.6 Contoh *Diagram Activity* Bank (Wikipedia.org)

## 5. Diagram Component dan Diagram Deployment

*Component* adalah sebuah *code module* (kode-kode modul). *Diagram Component* merupakan fisik sebenarnya dari *diagram Class*. *Diagram Deployment* menerangkan bahwa konfigurasi fisik *software* dan *hardware*.



Gambar 2.7 Contoh *Component Diagram* Dan *Deployment Diagram*  
(Wikipedia.org)

Fisik *hardware* berbentuk seperti *node-node*. Setiap komponen merupakan bagian dari *node*.

## **BAB III**

### **METODOLOGI PENELITIAN**

Pada bab ini akan dipaparkan langkah-langkah yang digunakan untuk membahas permasalahan yang diambil dalam penelitian. Dibagian ini juga dijelaskan alat dan metode yang digunakan untuk melakukan perencanaan dan mendapatkan spesifikasi kebutuhan pengguna, dan dibagian akhir dituliskan rencana pengerjaan skripsi.

Tahapan penelitian sesuai dengan metode yang dibuat, jika tidak sesuai maka perencanaan pembuatan sistem dan penelitian ini akan jauh dari harapan. Permulaan penelitian membutuhkan banyak informasi yang berkaitan tentang penelitian yang di butuhkan. Ketika akhir penelitian, maka sistem akan dilakukan pengujian yang sesuai dengan metode pengujian, guna memperoleh hasil pengujian yang maksimal.

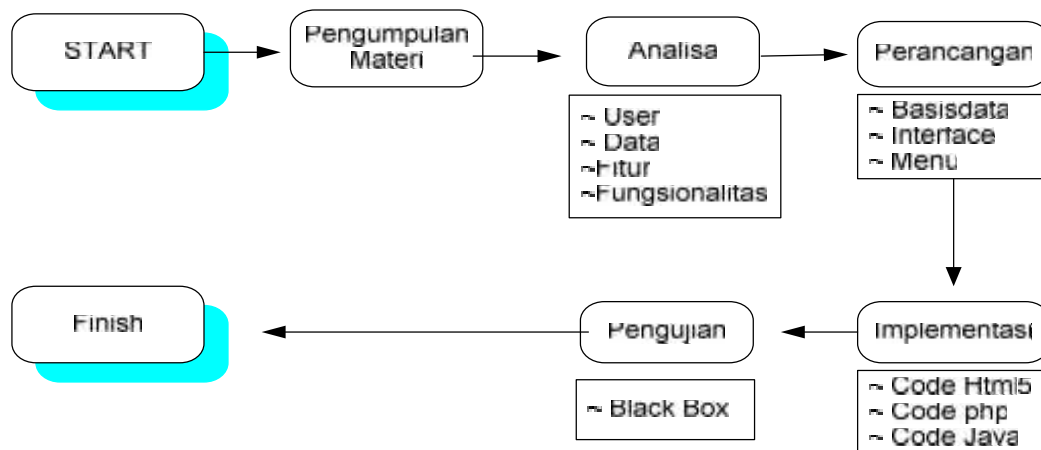
#### **3.1. Metode Pendekatan dan Pengembangan Sistem**

##### **3.1.1. Metode Pendekatan Sistem**

Metode pendekatan sistem yang di gunakan adalah pendekatan dengan Object Oriented yang menggunakan OOA(*Object Oriented Analisis*) dan OOD(*Object Oriented Design*) yang di visualisasikan dengan UML dan di antara nya adalah sebagai berikut : *Use Case, Activity Diagram, Sequence Diagram, Class Diagram, dan Deployment Diagram.*

##### **3.1.2. Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan adalah metode paradigma *Sekunsial Linier (Waterfall)* model, seperti yang tertera pada gambar berikut ini:



Gambar 3.1 Tahapan Penelitian

### 3.1.2.1. Pengumpulan Materi

Pengumpulan materi adalah langkah tahapan persiapan yang harus dilaksanakan terlebih dahulu sebelum dilakukan penelitian. Berikut adalah beberapa aktivitas yang dilaksanakan dalam pengumpulan materi.

#### 1. Rumusan Masalah

Rumusan masalah adalah proses identifikasi mengumpulkan dan mengklasifikasi masalah yang ada. Identifikasi ini sangat penting untuk kemudian menghasilkan solusi yang menjadi tujuan dari penelitian ini. Beberapa hal yang melatar belakangi penelitian ini adalah sebagai berikut:

- Perkembangan teknologi *mobile* terutama *smartphone* yang makin banyak digunakan oleh masyarakat membantu dalam menjalankan aktivitas.
- Gaya hidup *e-style* yang menunjang segala kegiatan dan membuat aktivitas menjadi lebih mudah dan efisien.
- Menjalankan program *go green* yang menjadi *trend center* dalam *information communication technology* (ICT), sehingga lebih kepada *less paper*
- Mobilitas manusia yang semakin tinggi untuk memanfaatkan teknologi *smartphone* dalam pemesanan menu kuliner pada sebuah restoran.

- e. Bagaimana merancang sebuah aplikasi *mobile-resto* kuliner dengan teknologi *multiplatform* berbasis *client server* sehingga lebih dinamis dan berguna bagi manusia.
- f. Persaingan bisnis bagi para pemilik restoran untuk menerapkan teknologi pada restoran mereka demi memperoleh pelanggan yang banyak.
- g. Dari beberapa latar belakang tersebut dapat disimpulkan tahapan perumusan masalah dalam penelitian ini adalah analisa kebutuhan untuk pemesanan menu makanan pada sebuah restoran, terutama pada perangkat *mobile* dengan *multiplatform*.

## 2. Studi Literatur

Fungsi dari studi literatur adalah untuk pendukung dari penelitian yang akan dilaksanakan. Teori-teori yang digunakan bersumber dari buku, jurnal dan penelitian-penelitian sejenis yang dapat mendukung pemecahan masalah dalam penelitian yang dilakukan. Literatur yang dikumpulkan antara lain ialah:

- a. Pemahaman tentang pembuatan sistem dengan html5 dan php
- b. Konsep dan pengertian dalam pembuatan database sistem.
- c. *JQueryMobile*, mencakup fungsi, kerja, dan penunjang untuk pembuatan pemrograman *mobile*.
- d. Sistem operasi *mobile* yaitu Android dan Blackberry.
- e. *Object Oriented Analysis and Design* (OOAD), mencakup deskripsi dan struktur OOAD.

### 3.1.2.2. Analisa dan Perancangan

#### 1. Analisa dan perancangan dengan permodelan UML

Proses ini akan menggunakan pendekatan OOAD dalam menentukan model dan struktur salah satu model yang digunakan dalam pengembangan sistem menggunakan metode *Object Oriented Analysis and Design* (OOAD) adalah UML. Tahapan yang dilakukan dalam permodelan UML meliputi *use case*



*diagram, activity diagram, class diagram, sequence diagram dan deployment diagram.*

## 2. Analisa dan perancangan basis data

Setelah menganalisa bagaimana sistem yang akan dibuat maka dilakukan analisa terhadap racangan basis data untuk melengkapi komponen sistem. Lokasi basis data akan berada pada *server*.

## 3. Perancangan antar muka (*User Interface*)

Perancangan *interface* system bertujuan untuk memudahkan komunikasi sistem dengan pengguna. Perancangan *interface* ditekankan pada tampilan yang baik, mudah dipahami dan digunakan serta adanya tombol-tombol yang familiar.

### 3.1.2.3. Implementasi

Setelah analisa perancangan rampung maka akan dilanjutkan pada tahapan implementasi. Pada tahapan ini merupakan pengkodean atau menerapkan *script coding*. Penulisan kode pada *client* menggunakan bahasa pemrograman html5 dengan *framework* *jQueryMobile*. Kemudian setelah itu dengan pemrograman java android. Sedangkan pengkodean pada *server* menggunakan html5, css3 dan PHP. Basis data sistem berada pada aplikasi phpmyadmin sebagai media yang membantu pembuatan basis data sistem aplikasi *mobile-resto* ini.

### 3.1.2.4. Pengujian Sistem

Pengujian sistem sangat dibutuhkan, ini bertujuan untuk mengetahui apakah aplikasi telah berjalan sesuai dengan kebutuhan. Standar pengujian yang dilakukan berkaitan dengan uji fitur dan *performance* selama sistem dijalankan. Model pengujian yang dilakukan adalah dengan pengujian *Black Box*. Pungujian ini dilakukan dengan cara *Acceptance Testing* yaitu untuk menguji apakah sistem yang dikembangkan sesuai dengan apa yang diharapkan berdasarkan yang tertuang dalam spesifikasi fungsi dari sistem tersebut.

Selanjutnya dilakukan *deploying* aplikasi untuk melihat kekurangan aplikasi oleh pembuat aplikasi, dan kemudian diambil kesimpulan tentang penggunaan aplikasi. Serta prosedur yang menunjang dalam pengujian. Mulai dari ketertarikan, fungsionalitas, efisien dan waktu.

### **3.1.3. Alat Bantu Analisis dan Perancangan**

Alat bantu analisis yang digunakan yaitu UML (*Unified Modeling Languages*) yang merupakan sebuah bahasa standar untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan *model* yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*.

UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu, diagram-diagram yang terdapat pada UML, diantaranya yaitu :

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Class Diagram*
5. *Deployment Diagram*

## **BAB IV**

### **ANALISA DAN PERANCANGAN**

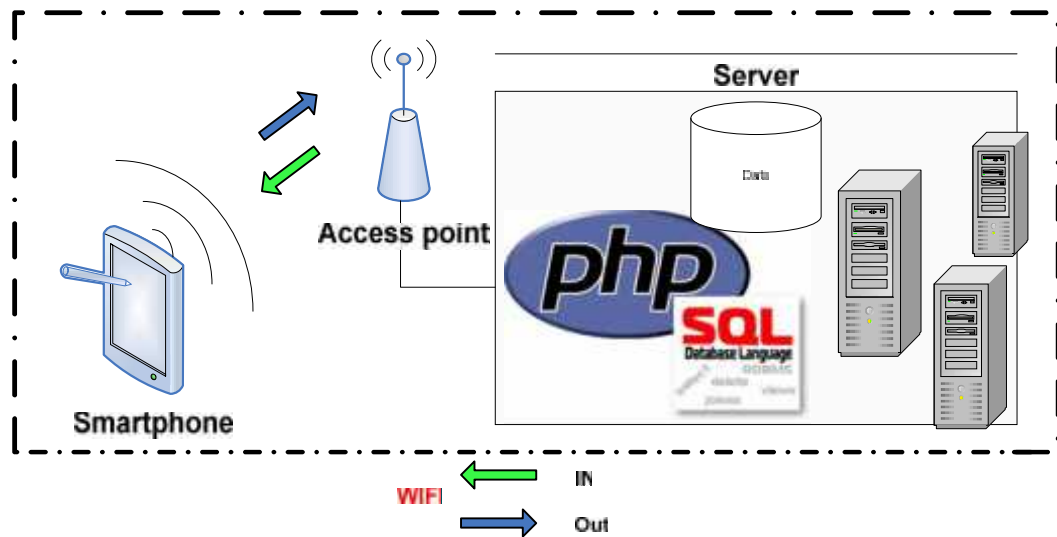
Pada bab ini akan dibahas mengenai analisa dan perancangan dari pembuatan sistem *mobile-resto reservation*. Pada bab sebelumnya telah dibahas hal yang berkaitan dalam pencarian informasi dan metodologi pembuatan sistem, dan pada bab ini adalah langkah selanjutnya dalam menganalisa sistem yang akan dibangun.

Analisa deskripsi dan arsitektur sistem menggunakan *Unified Modelling Language* atau yang disingkat dengan UML. UML merupakan rancangan dari struktur aplikasi yang akan dibangun. UML terdiri atas *Use Case*, *Class Diagram*, *Sequence Diagram* dan *Activity Diagram*.

#### **4.1. Gambaran Umum Sistem**

*Mobile-resto reservation* adalah sebuah teknologi baru yang akan membantu pelanggan dari sebuah rumah makan untuk melakukan pemesanan. Aplikasi *mobile-resto reservation* berbasis *client-server*. Aplikasi klien berjalan pada *smartphone* sedangkan pada sisi *server* berada pada sebuah komputer *server*. Bahasa pemrograman PHP (*Hypertext Preprocessing*) sebagai penghubung dalam pengiriman *request* dan penerimaan respon terhadap *server* yang menggunakan Apache dan *database mysql*.

Aplikasi *mobile-resto reservation* menggunakan jaringan WIFI (*Wireless fidelity*). Tidak menggunakan jaringan internet karena hanya dalam kawasan yang kecil. Aplikasi yang terdapat pada *smartphone* pelanggan harus terhubung kedalam jaringan wifi sehingga aplikasi dapat melakukan permintaan data menuju *server*, begitu sebaliknya *server* dapat merespon permintaan dari *client*.



Gambar 4.1. Arsitektur Sistem.

Dari gambar 4.1 diatas dapat dilihat proses kerja aplikasi yang akan dibuat, ada tiga bagian penting yang saling terhubung dalam kerja sistemnya, diantaranya:

1. Perangkat *smartphone* merupakan perangkat tempat berjalannya aplikasi *mobile-resto reservation*. Dari perangkat inilah pelanggan restoran melakukan pemesanan dengan sistem melalui PHP, dengan memanfaatkan jaringan WIFI yang tersedia pada *mobile smartphone* untuk mengakses *server*.
2. *Server*, terdiri dari dua bagian, yaitu:
  - a. *Connector* dengan menggunakan bahasa pemrograman PHP (*Hypertext Preprocessing*), berfungsi sebagai jembatan penghubung antara sistem yang berjalan pada perangkat Smartphone (*client*) dan *database*. Peran *connector* sangat penting, karena sisi *client* tidak bisa langsung Berinteraksi untuk meminta data dari *server* dan *database*. *Connector* ini yang bertugas mengirimkan *request* dan respon antara *client* dan *server*.
  - b. *Database*, merupakan bagian yang berfungsi sebagai *database* dari aplikasi *mobile-resto reservation*. *Database* ini yang bertanggung jawab

memberikan respon sesuai *request* dari *client*. *Database* yang digunakan adalah MySQL.

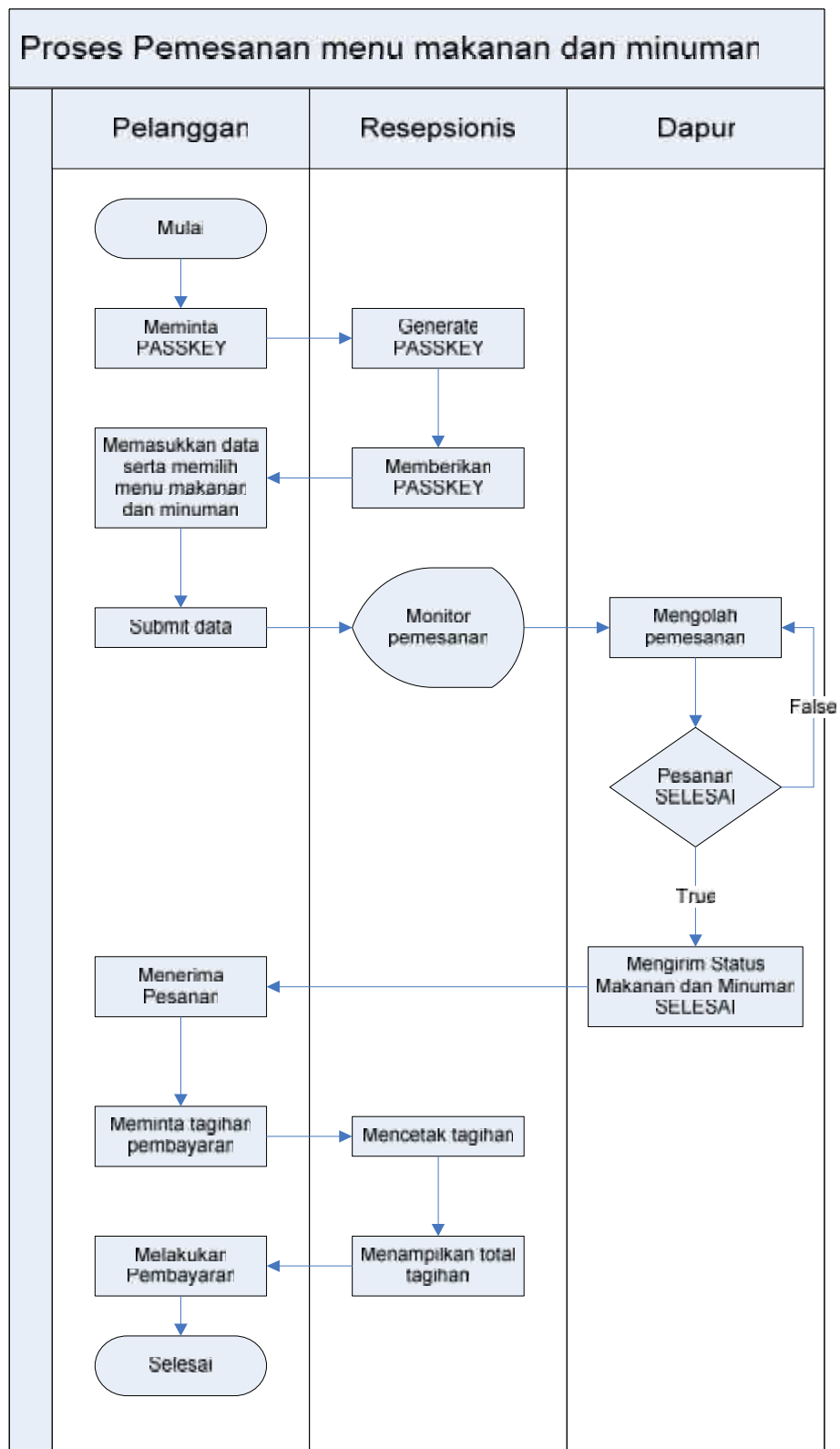
*User* atau pelanggan restoran langsung mendapatkan hak akses aplikasi jika aplikasi ini telah dipasangkan pada perangkat *Smartphone* pelanggan. Jika belum terpasang dapat dilakukan penginstalan terlebih dahulu.

Analisa awal sistem, pelanggan masuk ke restoran dan menerima *Passkey* dari resepsionis. *Passkey* digunakan untuk sekali pemesanan, jika kita tidak memiliki *passkey* kita tidak dapat memesan menu makanan di restoran tersebut. Ini dilakukan untuk menghindari penipuan data dari pihak luar restoran. Setelah pelanggan duduk di meja yang dipilih, maka tinggal menghubungkan *smartphone* ke WIFI dan memasukkan data serta *passkey* yang didapat, selanjutnya memilih menu makanan yang di pesan. Dengan batasan waktu pelanggan dapat membatalkan pesanan, setelah dilakukan proses masak maka pesanan tidak dapat dibatalkan. Dalam pemesanan terdapat sebuah catatan untuk pemesanan khusus.

Setelah itu pesanan disajikan, dan pelanggan menyantap makanan. Pelanggan yang sudah selesai makan maka menuju meja resepsionis untuk membayar tagihan. Resepsionis akan mencetak tagihan sesuai dengan data dan *passkey* yang diterima pelanggan.

#### **4.1.1. Proses Bisnis *Mobile-Resto Reservation***

Pihak restoran memiliki konsep bisnis untuk membuat usaha mereka menuju kepada kesuksesan yang lebih. Berbagai hal dilakukan untuk membantu dalam hal menunjang kesuksesan bisnis mereka. Aplikasi *mobile-resto reservation* adalah salah satu contoh sistem yang mendukung untuk kemajuan dan kesuksesan bisnis dari usaha restoran. Dengan adanya aplikasi ini membantu untuk mempermudah pelanggan memesan dan menjadi ketertarikan pelanggan. Dalam perancangan sistem ini membuat sebuah batasan waktu dalam pemesanan. Berikut bagan diagram proses pemesanan pada aplikasi *mobile-resto reservation* :



Gambar 4.2. Proses Alur Pemesanan.

#### 4.1.2. *Passkey*

Dalam rancangan *mobile-resto reservation*, digunakan sebuah *passkey* untuk sekali transaksi dalam satu waktu. *Passkey* diperoleh dari resepsionis. Untuk men-*generate passkey* dilakukan oleh sistem. Berikut beberapa alasan mengapa sistem *mobile-resto reservation* menggunakan *passkey* dalam transaksi pemesanan.

- a. Menghindari pemesanan palsu didalam restoran, jika tanpa adanya *passkey*.
- b. Menghindari adanya gangguan pemesanan palsu diluar restoran, dikarenakan signal WIFI masih terjangkau diluar restoran.

Rancangan *passkey* pada sistem *mobile-resto reservation* ini dengan menggunakan digit nomor angka, tidak huruf dan tidak gabungan huruf dan angka. Beberapa alasan mengapa dirancang tanpa gabungan huruf dengan angka atau huruf saja, karena lebih mempermudah user dalam menginputkan *passkey*. Jika dibuat dengan gabungan huruf dan angka, atau huruf saja, akan memperbanyak *digit* ruang pencarian dan pengetikan inputan oleh *user*. Maka dari itu penulis merancang *passkey* dengan beberapa *digit* nomor angka.

Berikut Algoritma penyusunan dan penggabungan *passkey* dengan angka.

**`$(nama_variabel_1) = random (00000 -> 99999)`**

**`$(nama_variabel_2) = random (99999 -> 00000)`**

**`$(nama_variabel_3) = $(nama_variabel_1). “” . $(nama_variabel_2)`**

Ketika angka nol muncul dalam pengacakan maka angka nol tersebut tidak akan tampil, sehingga minimal sebuah *passkey* akan muncul ketika diacak oleh sistem adalah 8 *digit passkey* sedangkan maksimal *passkey* yang akan muncul ketika diacak oleh sistem adalah 10 *digit passkey*.

## 4.2. Rancangan sistem

Aplikasi sistem *mobile-resto reservation* dibagi mejadi dua sisi perangkat, yaitu pada perangkat untuk *client* dan perangkat untuk *server*. *Device* yang digunakan untuk perangkat *client* adalah Smartphone. Pada bagian *server* menggunakan komputer *server* berbasis *web* untuk mengkonfigurasi hal-hal yang dibutuhkan.

### a. Sistem pada perangkat *server*

Kebutuhan sistem pada perangkat *server* :

1. Sebuah komputer *server* yang menampung *database* dan data sistem
2. Menggunakan bahasa pemrograman PHP sebagai penghubung.
3. Sebagai media untuk menjalankan program aplikasi *browser* dalam pengaturan dan konfigurasi yang dilakukan oleh Administrator
4. Sambungan langsung kepada perangkat WIFI sebagai penghubung peangkat *client*.
5. Memberikan sumberdaya dan merespon permintaan dari perangkat *client*.

### b. Sistem pada perangkat *client*

Kebutuhan sistem pada perangkat *client* :

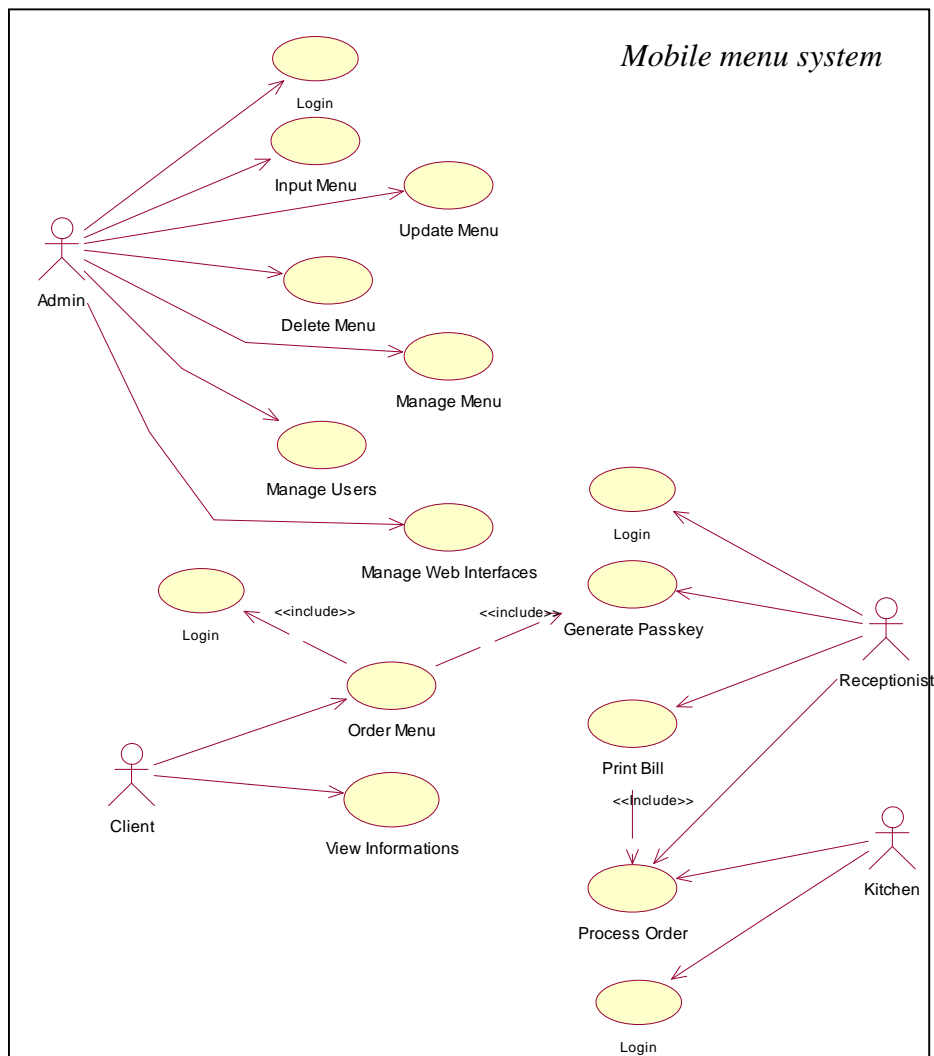
1. Pada *client* menggunakan perangkat cerdas yang disebut Smartphone.
2. Aplikasi harus telah ter-*install* dan dapat bekerja secara normal tanpa ada permasalahan
3. Menghubungkan perangkat *client* kepada koneksi WIFI.
4. Mendapatkan Passkey dari resepsionis untuk dapat melakukan *Submit* data.
5. Menggunakan bahasa pemrograman HTML5 dan framework JQueryMobile dalam membangun aplikasi *client*.
6. *Sourcecode* perangkat *client* dapat di-build kedalam berbagai sistem operasi mobile.



#### 4.2.1. Perancangan Model Sistem

Aplikasi *mobile-resto reservation* dirancang dengan menggunakan bahasa pemrograman berbasis objek *oriented*. Beberapa diagram yang akan dimodelkan menggambarkan rancangan sistem ini, yang terdiri dari *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram*, dan *deployment diagram*.

##### 4.2.1.1. Usecase Diagram



Gambar 4.3. Use Case Diagram.

#### 4.2.1.2. Usecase Diagram Spesification

Didalam sistem terdapat 4 aktor dan terdapat 11 *usecase*. Untuk menjelaskan *Use Case Diagram* dari aplikasi *mobile-resto reservation* dapat dilihat lebih jelas pada *Use Case Spesification*.

##### 1. Login

Tabel 4.1 dibawah ini adalah tabel *usecase specification login* yang dilakukan oleh aktor Admin, *kitchen*, *receptionist* dan *client* pada *usecase diagram*.

**Tabel 4.1 Usecase Spesification Login**

<i>Use Case</i>	<i>Login</i>
<i>Description</i>	Digunakan oleh semua akun untuk masuk kedalam sistem guna untuk dapat melakukan hal yang diperlukan.
<i>Actor</i>	Admin, <i>kitchen</i> , <i>receptionist</i> dan <i>client</i>
<i>Pre Condition</i>	Sistem menampilkan <i>form login</i>
<i>Post Condition</i>	Aktor berhasil melakukan <i>login</i> kedalam sistem
<i>Normal Flow</i>	Memasukkan data username dan password bagi akun admin, <i>kitchen</i> dan <i>receptionist</i> . Memeasukkan nama, no meja dan <i>passkey</i> bagi akun <i>client</i>

##### 2 Input Menu

Tabel 4.2 dibawah ini adalah tabel *usecase specification input menu* yang dilakukan oleh aktor Admin pada *usecase diagram*.

**Tabel 4.2 Usecase Spesification Input Menu**

<i>Use Case</i>	<i>Input Menu</i>
<i>Description</i>	Digunakan oleh aktor Admin untuk meng-input data makanan dan minuman kedalam sistem.
<i>Actor</i>	Admin
<i>Pre Condition</i>	Sistem menampilkan <i>form input menu</i>
<i>Post Condition</i>	Aktor berhasil memasukkan data makanan dan minuman kedalam sistem.
<i>Normal Flow</i>	Data yang dimasukkan Admin adalah berupa nama makanan atau minuman, deskripsi, harga dan gambar.

### *3 Update Menu*

Tabel 4.3 dibawah ini adalah tabel *usecase specification update menu* yang dilakukan oleh aktor Admin pada *usecase diagram*.

**Tabel 4.3 Usecase Spesification Update Menu**

<i>Use Case</i>	<i>Update Menu</i>
<i>Description</i>	Digunakan oleh aktor Admin untuk mengubah data makanan dan minuman didalam sistem.
<i>Actor</i>	Admin
<i>Pre Condition</i>	Sistem menampilkan <i>form update menu</i> .
<i>Post Condition</i>	Aktor berhasil mengubah dan menambah data baru makanan dan minuman kedalam sistem.
<i>Normal Flow</i>	Data yang diubah Admin adalah berupa nama makanan

	atau minuman, deskripsi, harga dan gambar.
<i>Alternative Flow</i>	Aktor Admin tidak merubah data apapun.

#### 4 Delete Menu

Tabel 4.4 dibawah ini adalah tabel *usecase specification delete menu* yang dilakukan oleh aktor Admin pada *usecase diagram*.

**Tabel 4.4 Usecase Spesification Delete Menu**

<i>Use Case</i>	<i>Delete Menu</i>
<i>Description</i>	Digunakan oleh aktor Admin untuk Menghapus data makanan dan minuman didalam sistem.
<i>Actor</i>	Admin
<i>Pre Condition</i>	Sistem menampilkan <i>form delete menu</i> .
<i>Post Condition</i>	Aktor berhasil Menghapus data makanan dan minuman didalam sistem.
<i>Normal Flow</i>	Data yang dihapus Admin adalah berupa nama makanan atau minuman, deskripsi, harga dan gambar.
<i>Alternative Flow</i>	Aktor Admin tidak menghapus data apapun.

#### 5 Manage Menu

Tabel 4.5 dibawah ini adalah tabel *usecase specification manage menu* yang dilakukan oleh aktor Admin pada *usecase diagram*.

**Tabel 4.5 Usecase Spesification Manage Menu**

<i>Use Case</i>	<i>Manage Menu</i>
-----------------	--------------------

<i>Description</i>	Digunakan oleh aktor Admin untuk mengelola menu makanan dan minuman didalam sistem.
<i>Actor</i>	Admin
<i>Pre Condition</i>	Sistem menampilkan <i>form</i> semua <i>menu</i> .
<i>Post Condition</i>	Aktor dapat mengelola data makanan dan minuman yang tersedia.
<i>Normal Flow</i>	Mengelola data makanan dan minuman, seperti berupa nama makanan atau minuman, deskripsi, harga dan gambar. Serta melakukan perubahan <i>status</i> makanan atau minuman.
<i>Alternative Flow</i>	Aktor Admin membutuhkan bagian dapur untuk mengelola <i>menu</i> makanan dan minuman.

## 6 Manage Web Interfaces

Tabel 4.6 dibawah ini adalah tabel *usecase specification manage web interfaces* yang dilakukan oleh aktor Admin pada *usecase diagram*.

**Tabel 4.6 Usecase Spesification Manage Web Interface**

<i>Use Case</i>	<i>Manage Web Interface</i>
<i>Description</i>	Digunakan oleh aktor Admin untuk mengelola tampilan utama <i>web</i> dan juga tampilan pada <i>client</i> .
<i>Actor</i>	Admin
<i>Pre Condition</i>	Sistem menampilkan halaman untuk mengelola tampilan utama dan klien.

<i>Post Condition</i>	Aktor berhasil mengubah dan menambah tampilan baru pada layar utama sistem dan pada klien.
<i>Normal Flow</i>	Mengelola tampilan <i>header</i> dan <i>footer</i> pada tampilan utama dan tampilan klien
<i>Alternative Flow</i>	Aktor Admin tidak merubah tampilan.

## 7 *Generate Passkey*

Tabel 4.7 dibawah ini adalah tabel *usecase specification generate passkey* yang dilakukan oleh aktor Receptionist pada *usecase diagram*.

**Tabel 4.7 Usecase Spesification Generate Passkey**

<i>Use Case</i>	<i>Generate Passkey</i>
<i>Description</i>	Digunakan oleh aktor Receptionist untuk men-generate sebuah kode khusus, syarat untuk meminta data.
<i>Actor</i>	Receptionist
<i>Pre Condition</i>	Sistem menampilkan halaman untuk men-generate sebuah kode.
<i>Post Condition</i>	Aktor berhasil men-generate sebuah kode.
<i>Normal Flow</i>	Menghasilkan sebuah kode khusus untuk pelanggan.
<i>Alternative Flow</i>	Jika <i>generate</i> gagal dapat dilakukan <i>regenerate</i>

## 8 *Print Bill*

Tabel 4.8 dibawah ini adalah tabel *usecase specification print bill* yang dilakukan oleh aktor Receptionist pada *usecase diagram*.

**Tabel 4.8 Usecase Spesification Print Bill**

<i>Use Case</i>	<i>Print Bill</i>
<i>Description</i>	Digunakan oleh aktor Receptionist untuk mencetak tagihan belanja pelanggan.
<i>Actor</i>	Receptionist
<i>Pre Condition</i>	Sistem menampilkan halaman untuk proses pembayaran.
<i>Post Condition</i>	Aktor berhasil menampilkan total bayar dan mencetak struk tagihan pelanggan.
<i>Normal Flow</i>	Menghasilkan sebuah struk tagihan.

#### 9 Process Order

Tabel 4.9 dibawah ini adalah tabel *usecase specification process order* yang dilakukan oleh aktor Receptionist dan kitchen pada *usecase diagram*.

**Tabel 4.9 Usecase Spesification Process Order**

<i>Use Case</i>	<i>Process Order</i>
<i>Description</i>	Digunakan oleh aktor Receptionist dan kitchen untuk melihat pemesanan yang masuk.
<i>Actor</i>	Receptionist dan Kitchen.
<i>Pre Condition</i>	Sistem menampilkan halaman monitor pemesanan dan halaman untuk mengelola pemesanan.
<i>Post Condition</i>	Aktor dapat melihat pemesanan yang masuk dan melakukan proses selanjutnya, serta dapat mengelola pemesanan.

<i>Normal Flow</i>	Berhasil memonitor dan mengelola proses pemesanan.
--------------------	--

#### 10 Order Menu

Tabel 4.10 dibawah ini adalah tabel *usecase specification order menu* yang dilakukan oleh aktor Client pada *usecase diagram*.

**Tabel 4.10 Usecase Spesification Order Menu**

<i>Use Case</i>	<i>Order Menu</i>
<i>Description</i>	Digunakan oleh aktor client untuk melakukan pemesanan menu makanan dan minuman.
<i>Actor</i>	Client.
<i>Pre Condition</i>	Sistem menampilkan halaman untuk memesan menu makanan dan minuman.
<i>Post Condition</i>	Aktor berhasil memesan menu makanan dan minuman.
<i>Normal Flow</i>	Berhasil men-submit data pemesanan ke <i>server</i> .
<i>Alternative Flow</i>	<i>Request time out</i>

#### 11 View Information

Tabel 4.11 dibawah ini adalah tabel *usecase specification view information* yang dilakukan oleh aktor Client pada *usecase diagram*.

**Tabel 4.11 Usecase Spesification View Information**

<i>Use Case</i>	<i>View Information</i>
<i>Description</i>	Digunakan oleh aktor client untuk Melihat berita dan artikel – artikel tentang informasi makanan dan minuman.

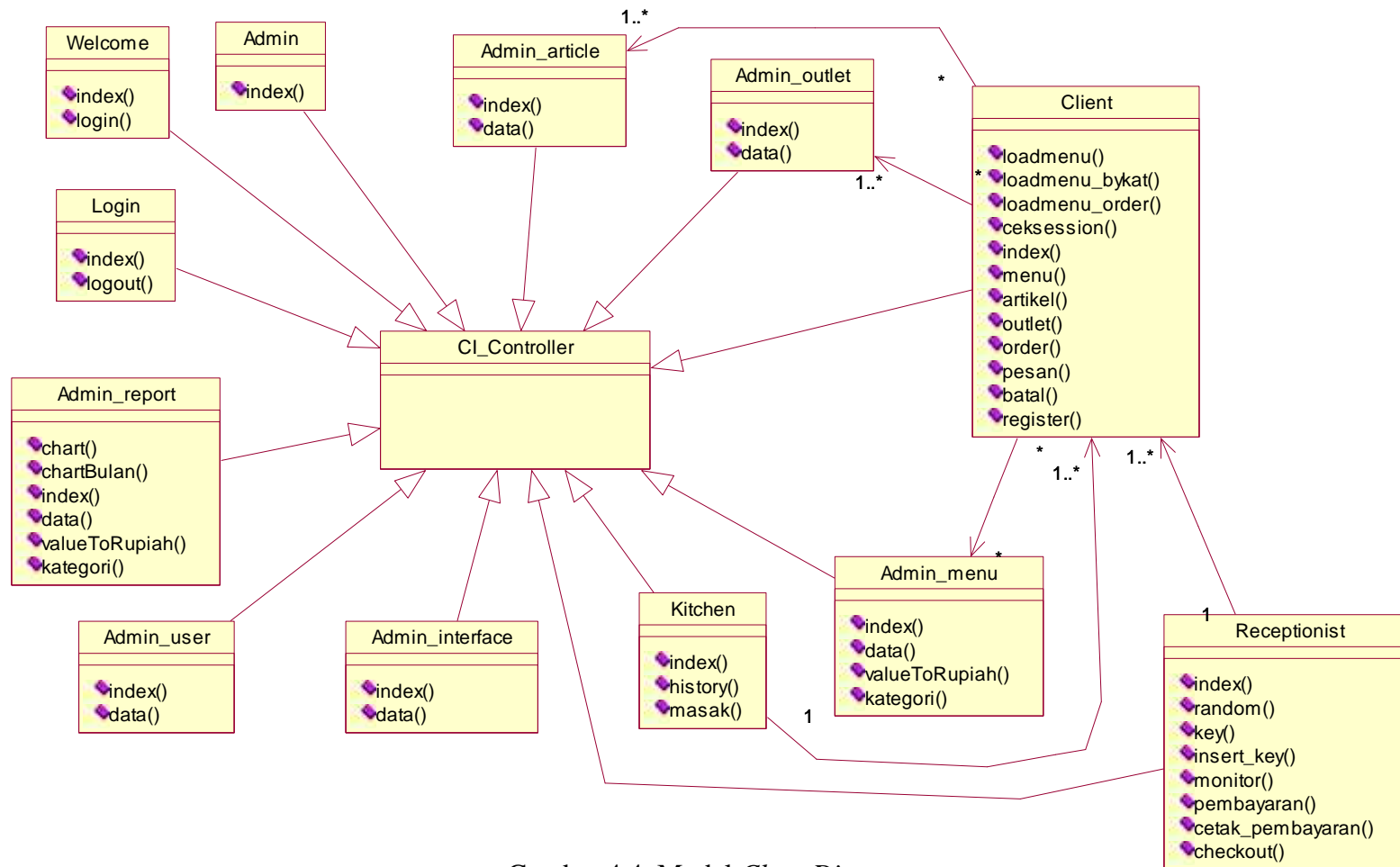


<i>Actor</i>	Client.
<i>Pre Condition</i>	Sistem menampilkan halaman berita dan artikel mengenai makanan dan minuman.
<i>Post Condition</i>	Aktor berhasil menampilkan halaman berita dan artikel.
<i>Normal Flow</i>	Berhasil menampilkan halaman berita dan artikel.

#### 4.2.1.3. *Class Diagram*

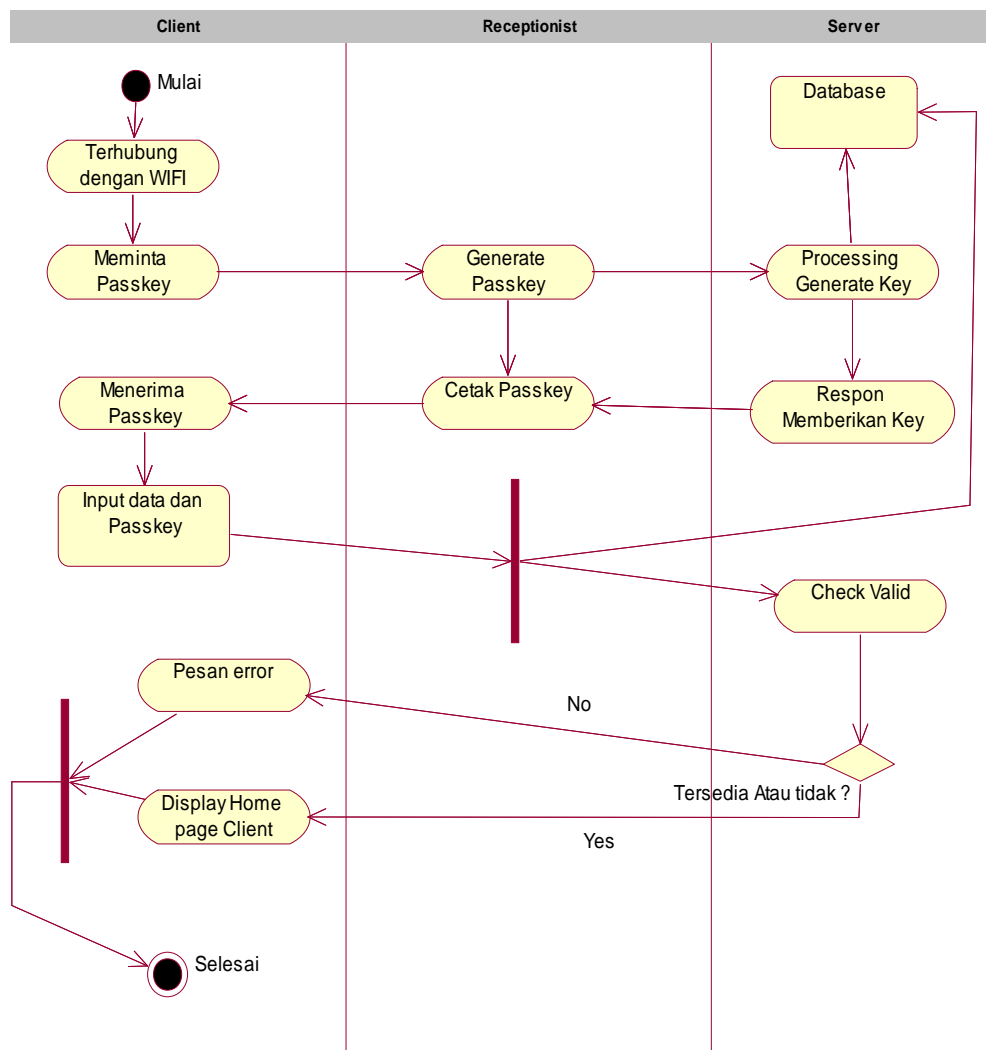
*Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

*Class Diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. *Class diagram* yang dijelaskan pada analisa ini adalah *class diagram* yang terpasang pada sistem *mobile-resto reservation*. *Mobile-resto reservation* menggunakan *framework* codeigniter, maka terdapat sebuah *class* utama sebagai *class* pada *framework* itu sendiri. *Class extend* adalah *class* yang dirancang dalam pembuatan aplikasi *mobile-resto reservation*. Gambar 4.4 dibawah ini menjelaskan *class diagram* sistem aplikasi *mobile-resto reservation*.



#### 4.2.1.4. Activity Diagram

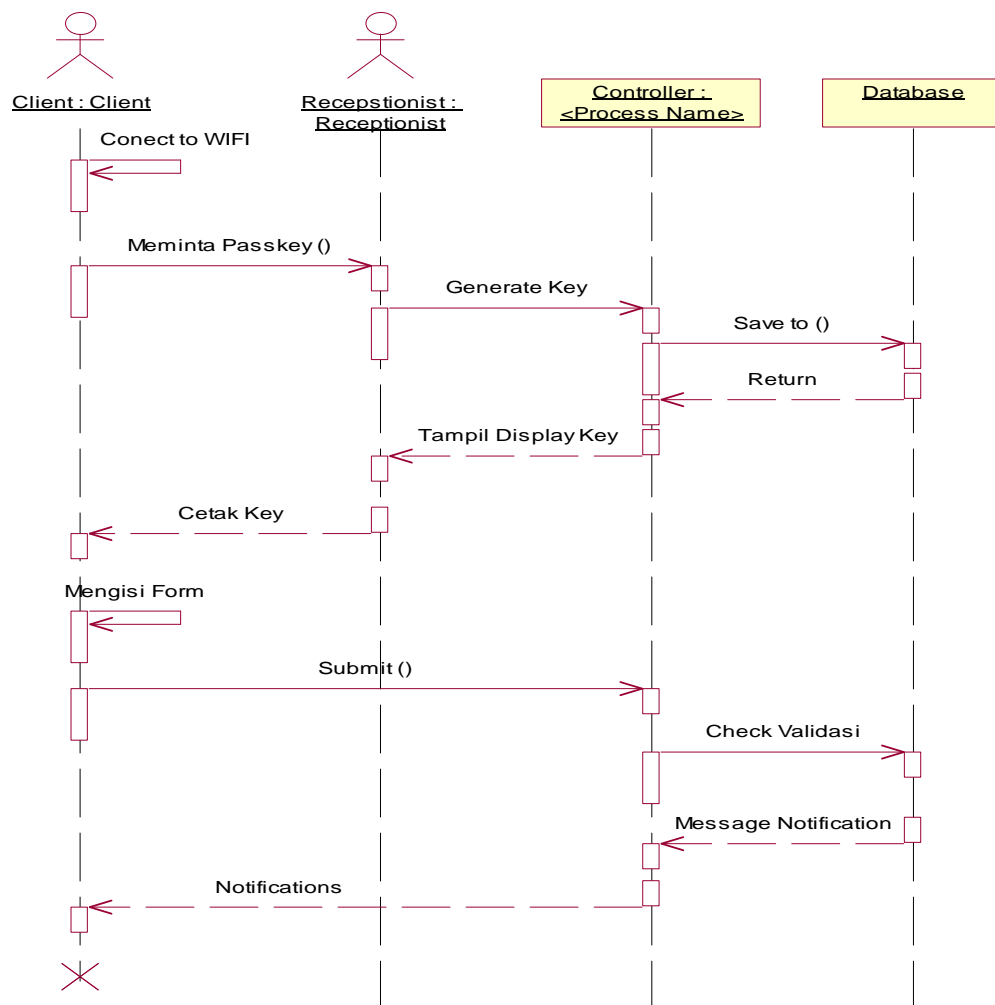
*Activity diagram* menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti. *Activity diagram* juga dapat menggambarkan proses *parallel* yang mungkin terjadi pada beberapa eksekusi. Menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing – masing alur berawal, keputusan yang mungkin terjadi, dan bagaimana mereka akan berakhir. Gambar 4.5 dibawah ini menggambarkan *activity diagram* untuk proses *login* pada akun *client*. Untuk *activity diagram* lainnya dapat di lihat pada lampiran A.



Gambar 4.5. Activity Diagram Login Client.

#### 4.2.1.5. Sequence Diagram

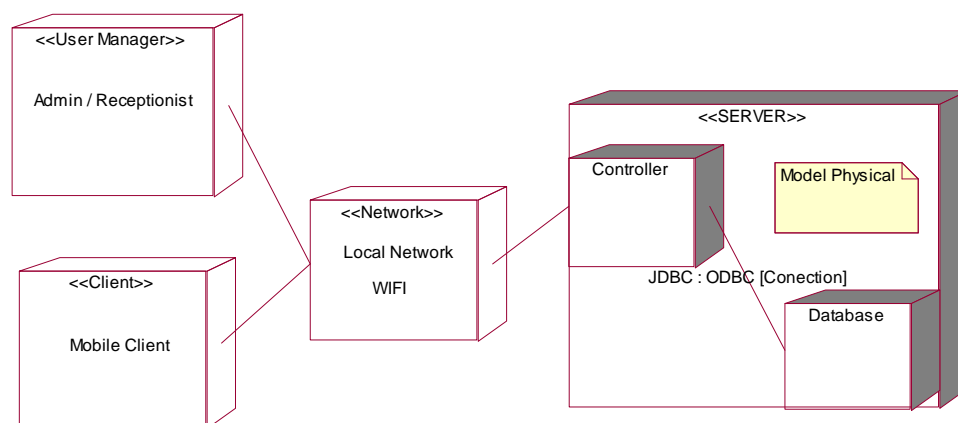
Sequence diagram merupakan interaksi antara objek-objek dalam suatu sistem dan terjadi komunikasi yang berupa pesan (*message*) serta parameter waktu, dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram. Berikut Gambar 4.6 adalah *sequence diagram* untuk proses login pada akun *client*. Untuk *sequence diagram* lainnya dapat dilihat pada lampiran A.



Gambar 4.6. Sequence Diagram Login Client.

#### 4.2.1.6. Deployment Diagram

*Deployment diagram* adalah pandangan yang terkait dengan penyebaran fisik aplikasi. Hal ini termasuk persoalan layout jaringan dan lokasi komponen-komponen dalam jaringan. *Deployment* berisikan prosesor-prosesor, peralatan-peralatan, proses-proses dan hubungan antar prosesor dan antar peralatan. Semua informasi ini digambarkan dalam suatu *deployment diagram*. Hanya ada satu *deployment diagram* dalam setiap sistem, sehingga hanya satu *deployment* dalam setiap model. Suatu *deployment diagram* menampilkan semua titik (*node*) dalam suatu jaringan, hubungan antar mereka, dan proses-proses yang dijalankan pada masing-masing *node*. Berikut ini adalah gambar untuk *deployment diagram* pada *mobile-resto reservation*.



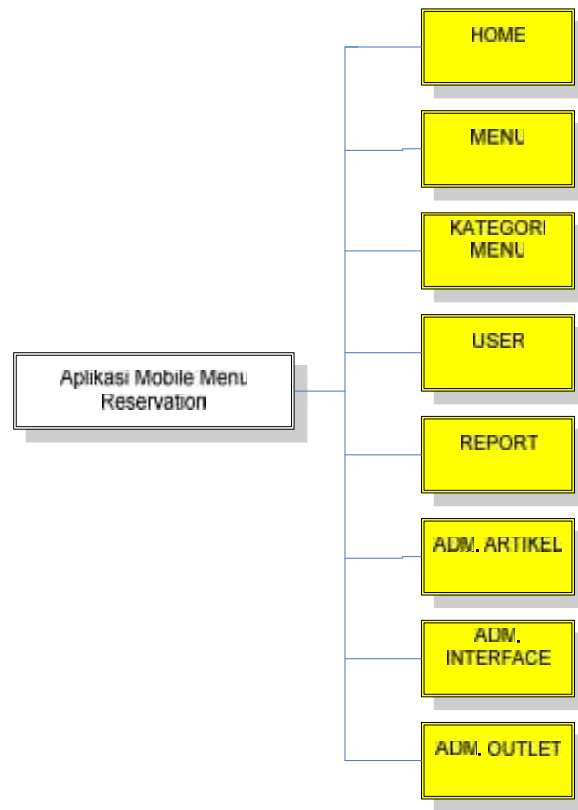
Gambar 4.7. *Deployment Diagram*.

#### 4.2.2. Perancangan Struktur Menu

Menu merupakan pilihan untuk menunjukan langkah *activity* dari *user*, dalam sistem aplikasi *mobile-resto reservation* ini memiliki beberapa tampilan menu utama dari sisi akun client dan akun lainnya. Masing – masing akun memiliki tampilan *menu* yang berbeda untuk setiap aktivitasnya. Berikut struktur *menu* untuk beberapa kategori akun dari sistem aplikasi *mobile-resto reservation*.

## 1. Admin

Menu pada tampilan admin :

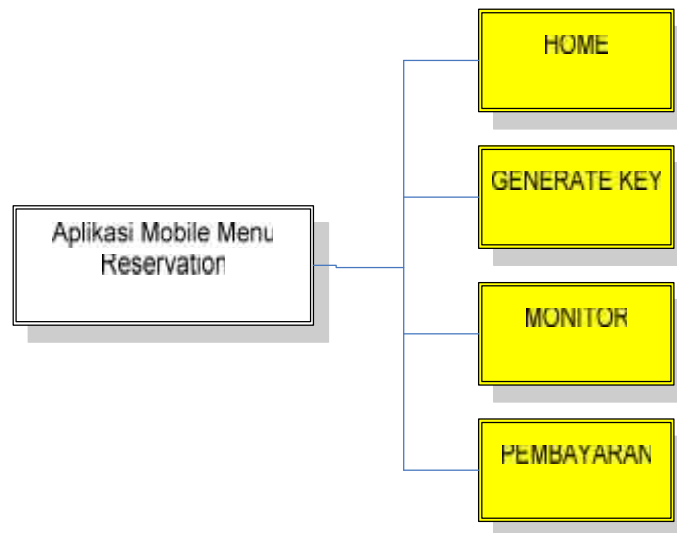


Gambar 4.8. Tampilan Struktur Menu Akun Admin.

Gambar 4.8 menampilkan struktur *menu* pada akun Admin. Terdapat 8 *sub menu* untuk membantu aktivitas dari akun admin, serta fungsi yang dibangun dalam akun tersebut. *Sub menu* dengan nama “Menu” adalah untuk mengelola *menu* makanan dan minuman. *Sub menu* “Kategori menu” adalah untuk mengelola kategori yang ada didalam makanan dan minuman sebuah restoran. *Sub menu* “User” adalah untuk mengelola *User* yang tersedia didalam sistem. *Sub menu* “Report” adalah untuk melihat hasil *report data* penjualan. *Sub menu* “ADM Artikel” adalah untuk mengelola artikel yang ditampilkan di-*Client*. *Sub menu* “ADM Outlet” adalah untuk mengelola informasi *Outlet* yang tersedia.

## 2. Resepsionis

Menu pada tampilan Resepsionis :

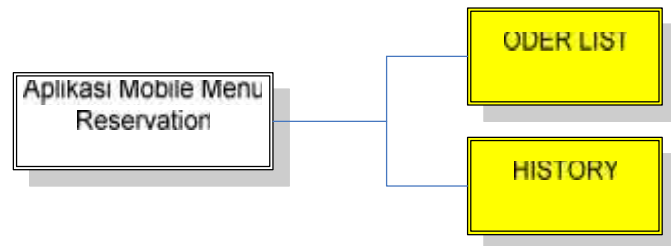


Gambar 4.9. Struktur *Menu* Pada Akun Resepsionis.

Gambar 4.9 Merupakan tampilan *menu* dari akun Resepsionis. Terdapat 4 *sub menu*, untuk membantu aktivitas dari akun Resepsionis. *Sub menu* “*Home*” Merupakan beranda dari akun resepsionis. *Sub menu* “*Generate Key*” adalah untuk men-*generate key* yang dibutuhkan untuk *client*. *Sub menu* “*Monitor*” adalah untuk me-*monitor client* yang masuk dan memesan makanan dan minuman. *Sub menu* “*Pembayaran*” adalah untuk melakukan proses setelah *client* selesai makan dan minum.

### 3. Dapur

Menu pada tampilan Dapur :

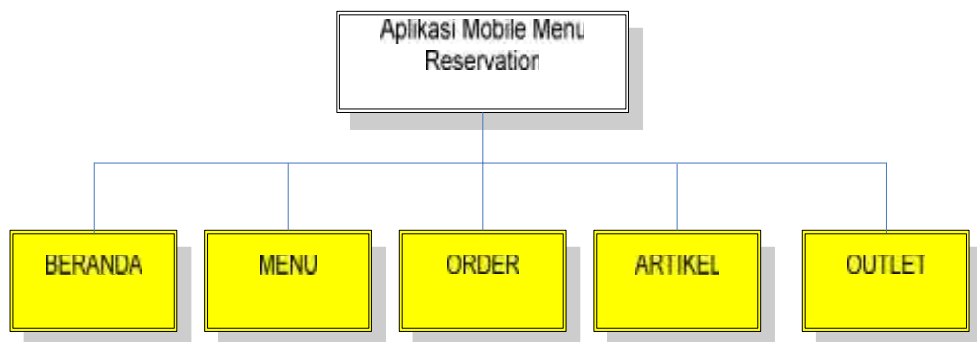


Gambar 4.10. Struktur Menu Pada Akun Dapur.

Gambar 4.10 merupakan *menu* dari akun *kitchen*. Terdapat dua *sub menu* yaitu *order list* dan *history*. *Sub menu order list* adalah untuk melakukan proses masak dan lainnya. *Sub menu history* adalah untuk melihat proses *history* yang telah diproses.

### 4. Klien

Menu pada tampilan klien :



Gambar 4.11. Struktur Menu Pada Akun *Client*.

Gambar 4.11 merupakan tampilan menu dari akun klien. Terdapat 4 *sub menu*, yaitu *menu*, *order*, *artikel* dan *outlet*. *Sub menu "Menu"* adalah untuk memilih *menu* makanan dan minuman. *Sub menu "Order"* adalah untuk melihat makanan dan minuman yang telah dipesan. *Sub menu "Artikel"* adalah untuk



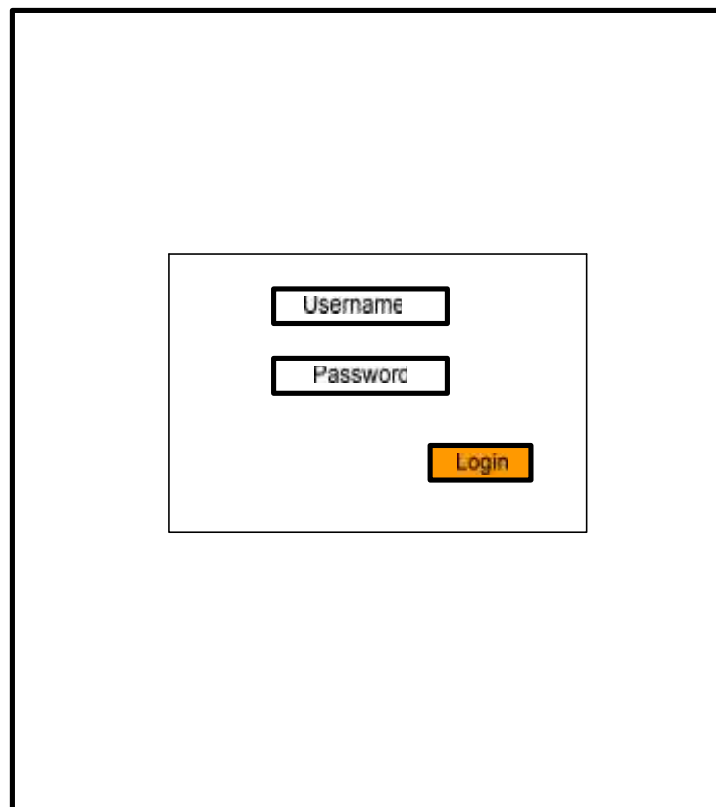
melihat artikel – artikel yang di *posting* oleh admin. *Sub menu “Outlet”* adalah untuk melihat *outlet – outlet* lain yang dimiliki oleh restoran tersebut.

#### 4.2.3. Perancangan Antarmuka Sistem

Perancangan antarmuka sistem sangat diperlukan, *interface* atau antarmuka sistem *mobile-resto reservation* terbagi menjadi dua bagian antarmuka. Antarmuka yang pertama untuk tampilan diakun admin, resepsionist dan dapur *via desktop*. Antarmuka yang kedua *via mobile* untuk akun klien.

##### 4.2.3.1. Tampilan *Login* Dekstop

Tampilan *login* untuk akun admin, resepsionist dan dapur dengan tampilan *via desktop*.

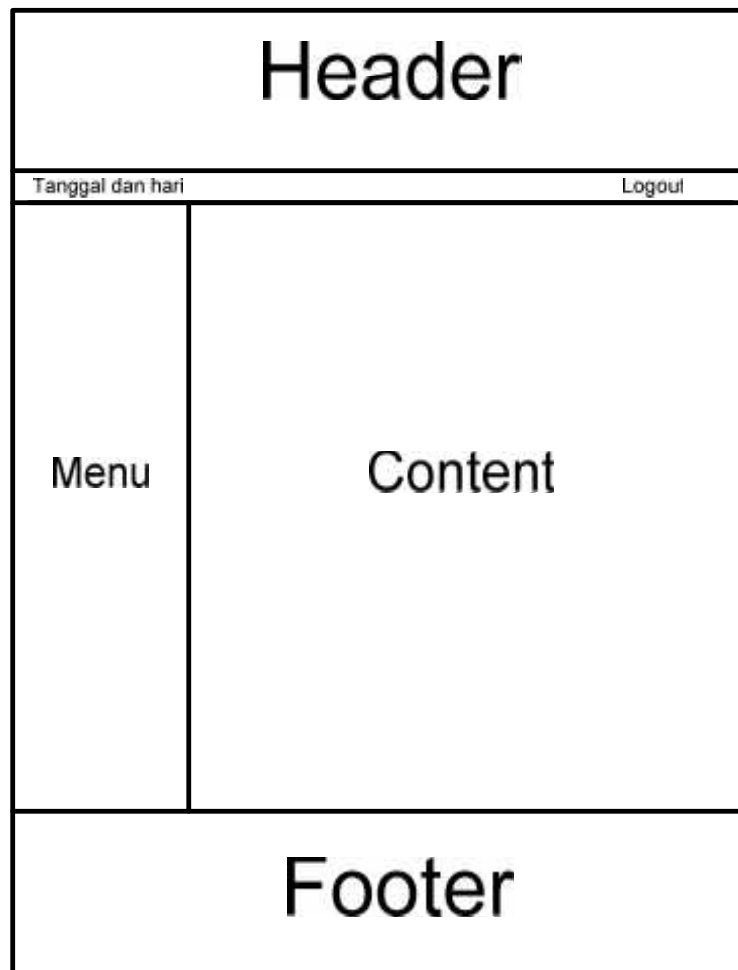
The diagram illustrates a desktop login interface. It consists of a large outer rectangle representing the desktop window. Inside this window, there is a smaller rectangle representing the login form. Within the login form, there are three elements: a text input field labeled 'Username', a text input field labeled 'Password', and a button labeled 'Login'. The 'Login' button is highlighted with a yellow background and a black border.

Gambar 4.12. Rancangan *Login* Via Dekstop.

Untuk tampilan awal *login* hanya menampilkan *field* untuk *username* dan *password*. Pada bagian bawah ada *button* untuk *login*.

#### 4.2.3.2. Tampilan *Content* Dekstop

Tampilan *content* merupakan antarmuka dari semua fitur *menu* yang ada pada akun admin, resepsionis dan dapur. Berikut tampilan *content* untuk akun admin, resepsionist dan dapur.

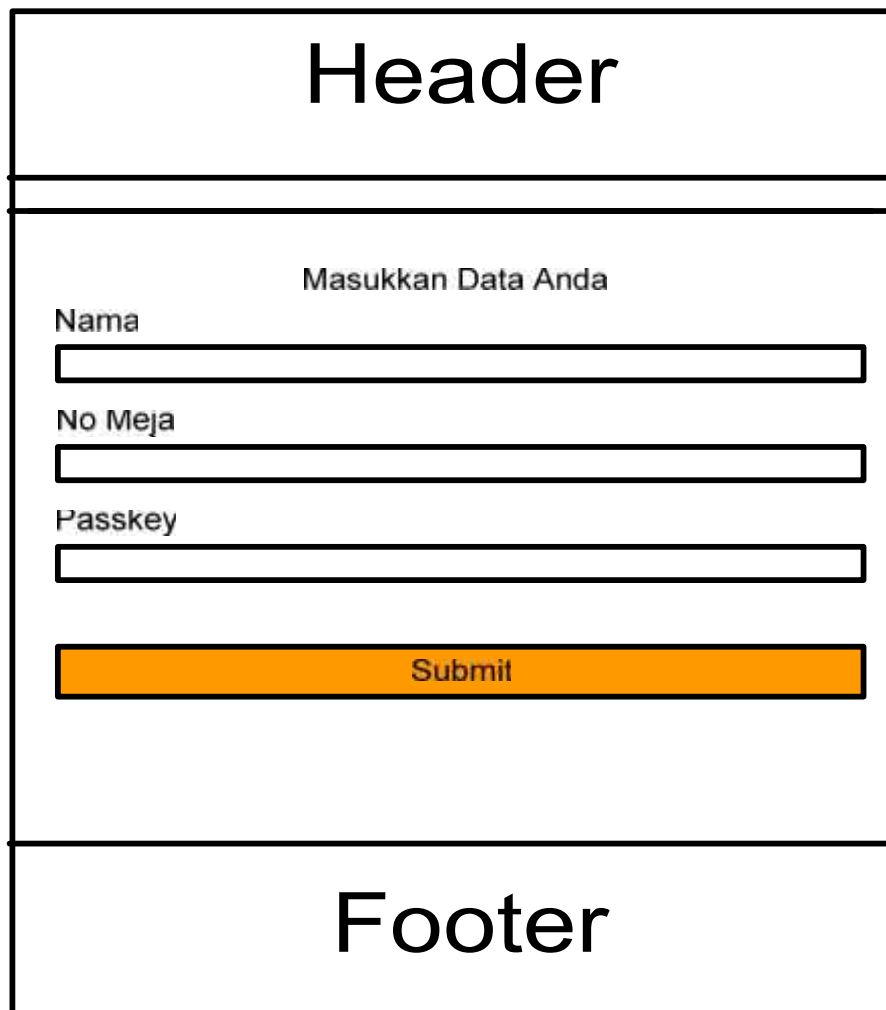


Gambar 4.13. Rancangan Antarmuka *Content* Via Dekstop.

Gambar 4.13 adalah Antarmuka apa yang terdapat didalam akun admin, resepsionist dan dapur. Pada rancangan *content* berada pada bagian kanan. Pada bagian atas terdapat *header* yang menjadi tampilan awal dari sistem. Pada bagian kiri adalah *menu – menu*, dan yang dibagian bawah terdapat *footer*.

#### 4.2.3.3. Tampilan *Login Client*

Antarmuka untuk login sebelum memean pada akun klien, berikut gambar rancangan yang akan dibangun.



The wireframe illustrates the layout of the Client Login interface. It is structured into three main horizontal sections: a top header, a central content area, and a bottom footer. The header section contains the word 'Header' in a large, bold font. The central content area is titled 'Masukkan Data Anda' (Enter Your Data) and contains three input fields labeled 'Nama' (Name), 'No Meja' (Table Number), and 'Passkey'. Below these fields is a prominent orange 'Submit' button. The footer section contains the word 'Footer' in a large, bold font.

Gambar 4.14. Rancangan Antarmuka *Login Client*.

Pada rancangan tampilan *login* klien, terdapat 3 *field* untuk melakukan pengisian. Pertama adalah Nama, merupakan nama orang yang memesan menu makanan dan minuman. Kedua adalah No meja, di meja mana Pelanggan akan duduk. Selanjutnya yang ketiga adalah *Passkey*, merupakan syarat utama yang harus ada sebelum memesan.

#### 4.2.3.4. Tampilan *Content Client*

Setelah melakukan *login* untuk akun klien, maka masuk pada tampilan *content* dari akun klien. Berikut adalah rancangan tampilan *content* untuk akun klien.

Header	Menu	Order	Artikel	Outlet
<div>Search menu <input type="text"/></div>				
<div>Image <input type="text"/></div>				
<div>Image <input type="text"/></div>				
<div>Image <input type="text"/></div>				
<div>Image <input type="text"/></div>				
<div>Image <input type="text"/></div>				
<div></div>				
Footer				

Gambar 4.15. Rancangan Antarmuka *Content Client*.

Gambar 4.15 Merupakan rancangan antarmuka *content* dari akun klien, terdapat *menu - menu* yang mempermudah pelanggan untuk melakukan pemesanan makanan dan minuman.

## **BAB V**

### **IMPLEMENTASI DAN PENGUJIAN**

Pada bab ini adalah menjelaskan tentang implementasi dan pengujian sistem, Pada bab IV analisa dan perancangan menjelaskan perancangan sistem yang akan dibangun. Setelah dirancang maka dilakukan *coding* untuk menerapkan *code-code* kedalam sistem yang akan dibangun. Selanjutnya dilakukan implementasi dan pengujian pada sistem yang dibangun.

Tahapan implementasi merupakan tahapan dimana sistem informasi yang telah dirancang, dianalisa, dan dibangun, lalu diuji kelayakannya untuk selanjutnya dioperasikan sebagaimana mestinya sesuai dengan fungsi dan kelayakannya. Berikut ini akan dijelaskan tentang pengimplementasian dari analisis dan perancangan yang telah dilakukan terhadap aplikasi *mobile-resto reservation*.

#### **5.1. Batasan Implementasi**

Berikut dijelaskan batasan implementasi aplikasi *mobile-resto reservation*:

- a. Aplikasi berada pada *server* utama yang terhubung dengan WIFI.
- b. Terdapat 4 akun, untuk menggunakan aplikasi *mobile-resto reservation*.  
Yaitu : klien, Admin, Resepsionis dan kitchen (dapur) .
- c. Pengujian untuk aplikasi *client* hanya pada *Smartphone* Android dan Blackberry. Aplikasi ini mencakup fitur- fitur sebagai berikut:
  - Penyajian informasi mengenai menu makanan dari restoran, berupa nama makanan, deskripsi, harga, dan gambar.
  - Aplikasi menggunakan sistem *Passkey*.

- Pelanggan dapat membatalkan pesanan jika belum dimasak
  - Terdapat sebuah catatan pemesanan pada aplikasi *client*
  - Pemesanan memiliki batas waktu.
- d. Aplikasi untuk *administrator* hanya dapat diakses melalui *web*. Secara umum aplikasi ini memiliki fitur-fitur sebagai berikut.
- *Administrator* dapat menginput data makanan dan minuman, harga dan gambar.
  - Mengelola kategori *menu*
  - Mengelola artikel yang ditampilkan pada aplikasi *Client*.
  - Mengelola *outlet* yang ditampilkan pada aplikasi *Client*
  - Mengelola *interface header* dan *footer* Restoran.
  - Melihat *report* penjualan Per-hari
- e. Untuk akun resepsionis digunakan oleh resepsionis sendiri, berikut fitur yang bisa digunakan.
- *Generate Passkey*
  - Mencetak *Passkey*
  - Validasi pembayaran tagihan
  - Cetak struk.
  - Monitor Pemesanan.
- f. Akun ke empat bagian dapur hanya dapat melihat pemesanan klien yang telah *login* dengan *Passkey*.

## 5.2. Lingkungan Operasional

Komponen-komponen yang dibutuhkan untuk menerapkan aplikasi ini antara lain berupa komponen *hardware* dan *software*. Seperti yang telah dijelaskan pada analisa kebutuhan sebelumnya, maka berikut ini adalah lingkungan operasional yang merupakan lingkungan tempat aplikasi ini diimplementasikan dan digunakan oleh penulis dengan spesifikasi sebagai berikut:

1. Perangkat keras

Processor : *Pentium(R) Dual-Core CPU T4400 @ 2.20GHz*  
Memori (RAM) : 2.00 GB  
Access point : Access point TP-Link TL-wn322g

2. Perangkat Lunak

Sistem Operasi : *Windows 7 Ultimate 32-bit Operating System*  
Bahasa Pemrograman : Java, PHP, XML  
*Tools Pengembangan* : Eclipse Juno,  
: *Java Development Kit 7u9 (JDK 7u9)*  
: *Android SDK, ADT 21.1*  
: *Android Virtual Device*  
: *Blackberry WebWorkSDK*  
: *Blackberry Simulator*  
*Tools Perancangan* : Dreamweaver, dan Notepad ++  
*Server* : Xampp (apache, mysql, phpmyadmin)  
Pemodelan UML : Rational Rose Enterprise Edision

### 5.3. Lingkungan Implementasi

Dalam Lingkungan implementas, aplikasi klien dibagi menjadi dua kategori, yaitu pada Android dan Blackberry. Lingkungan implementasi aplikasi ini dilakukan dengan menggunakan perangkat keras dan perangkat lunak diantaranya:

1. Android

- Perangkat keras : *Smartphone* Android Samsung Galaxy Chat
- Perangkat lunak : Sistem operasi Android 4.0.1 (froyo)


2. Blackberry

- Perangkat keras : *Smartphone* Blackberry 9780 (Onyx 2)
- Perangkat lunak : Sistem operasi Blackberry OS 6.0

### 5.3.1. Instalasi Aplikasi Pada *Smartphone*

Pada saat melakukan pengujian pada aplikasi *client*, terlebih dahulu program aplikasi *client* harus telah dibuat *master installer*. Master installer ini yang akan dipasang pada masing – masing *smartphone* yang akan dilakukan pengujiannya. Master installer untuk *smartphone* android ber-extension kan dengan .apk sedangkan untuk blackberry ber-extension kan .cod atau .jad

Berikut *icon* dan nama aplikasi *mobile-resto reservation* pada *smartphone client*.

Icon : 

Nama : Daily Menu

Setelah dilakukan pemasangan pada setiap *smartphone*, maka aplikasi dapat diuji berhasil atau tidaknya bekerja. Berikut adalah gambar yang menampilkan pengujian pada *smartphone* android yang telah berhasil dipasang dan berfungsi dengan baik.



Gambar 5.1. *Icon* Setelah Instalasi Sukses (Android).



Telihat *icon* telah muncul pada semua *menu* yang ada di *smartphone*. Menandakan instalasi selesai dan berhasil.



Gambar 5.2. Tampilan *Login* Pada Aplikasi Klien (Android).

Gambar 5.2 menampilkan halaman *login* sebelum dimasukkannya data pelanggan. Setelah data dimasukkan, pelanggan dapat melakukan pemesanan. Selanjutnya Menampilkan gambar pada blackberry saat instalasi berhasil dipasang.



Gambar 5.3. *Icon* Setelah Instalasi Sukses (Blackberry).

Gambar 5.3 menampilkan icon aplikasi *mobile-resto reservation* telah berhasil dipasang pada perangkat *smartphone* blackberry. Berikut tampilan layar utama sebelum dilakukan inputan data.



Gambar 5.4. Tampilan *Login* Pada Aplikasi Klien (Blackberry).

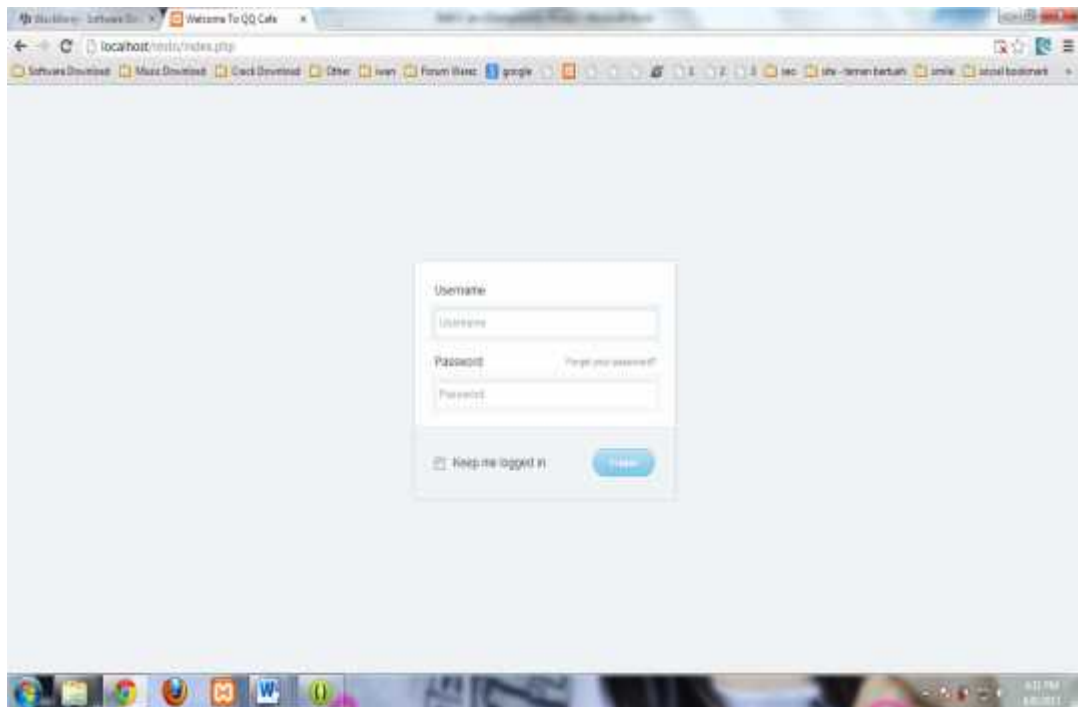
Sedikit berbeda dengan tampilan *login* pada *smartphone* android yang terlihat hampir keseluruhan, tampilan pada blackberry kecil sehingga harus diturunkan *view* untuk melihat keseluruhan tampilan. Untuk melihat tampilan lainnya dilampirkan pada lampiran B.

### 5.3.2. *Interface Pada Sisi Server*

Pada bagian server yang terletak pada sebuah *personal computer* dan dijalankan dengan *service* apache untuk memberikan layanan kepada klien yaitu pada klien android atau blackberry.

Pada aplikasi sisi *server* tidak perlu dilakukan penginstalan, melainkan langsung menuju pada implementasi aplikasi secara keseluruhan. Langkah awal ialah menghubungkan *server* dengan *access point* untuk memancarkan *signal wifi*. Kemudian menjalankan layanan apache dan *Mysql server* untuk dapat diakses oleh perangkat klien.

Berikut tampilan aplikasi pada bagian sisi *server* khusus untuk akun admin, resepsionis dan bagian dapur.



Gambar 5.5. Tampilan *Login* Pada Sisi *Server*.

Gambar 5.5 menampilkan tampilan sebelum melakukan *login* pada setiap akun terkecuali pada akun klien. Untuk melihat tampilan lainnya terdapat pada lampiran B.

#### **5.4. Tahapan Pengujian**




Tahapan pengujian dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang diharapkan. Pada tahap pengujian sistem ini perangkat keras yang digunakan yaitu *smartphone* Android dengan sistem operasi 4.0.1 (ICS) dan Blackberry 9780 OS 6.0

#### **5.4.1. Pengujian *Blackbox Mobile-Resto Reservation***

Aplikasi *mobile-resto reservation* memerlukan pengujian *blackbox* untuk melihat hasil kineja serta fitur fungsionalitas dari keseluruhan sistem. Pengujian dilakukan untuk memeriksa kekompakan atau kinerja antar komponen sistem yang diimplementasikan. Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Tujuan utama dari pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.. Adapun hasil dari pengujian ini dapat dilihat pada tabel dibawah ini.

Table 5.1. Pengujian aplikasi pada *smartphone* Android 4.0.1 dengan metode *Blackbox*




NO	Nama Pengujian	Deskripsi	Device	Pre kondisi	Prosedur Pengujian	Masukan	Hasil yang diharapkan	Hasil yang didapat	Ket
1	Membuka aplikasi	Menjalankan Aplikasi yang telah dipasang di <i>smartphone</i>	Android 4.0.1	-	Klik <i>Icon Daily Menu</i>	-	Menampilkan halaman <i>login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	Tampilan <i>form login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	<b>Berhasil</b>
2	Melakukan <i>login</i>	Pengujian untuk masuk kehalaman menu makanan dan minuman dengan <i>login</i> terlebih dahulu	Android 4.0.1	Halaman <i>Login</i>	Mengisi <i>form login</i>	Masukkan <i>username, password</i> dan <i>passkey</i>	Masuk ke tampilan halaman <i>menu Makanan dan minuman</i>	Tampilan halaman menu makanan dan minuman	<b>Berhasil</b>
3	Memesan makanan atau minuman	Melakukan pemesanan makanan atau minuman	Android 4.0.1	Halaman <i>menu makanan dan minuman</i>	Memilih makanan atau minuman dengan klik salah satunya.	Klik makanan atau minuman yang akan di pilih	Makanan atau minuman yang dipilih tampil pada <i>tab order</i>	Makanan atau minuman yang dipilih berhasil	<b>Berhasil</b>
4	Melihat Pemesanan	Melihat daftar menu yang telah	Android 4.0.1	Halaman Menu	Klik <i>tab order</i>	-	Menampilkan daftar <i>list</i> yang telah	Tampilan <i>list</i> yang telah dipesan	<b>Berhasil</b>

		dipesan					dipesan sebelumnya		
5	Melihat total harga pesanan	Nilai total harga yang telah dipesan	Android 4.0.1	-	Klik <i>tab order</i> atau klik <i>tab</i> lainnya	-	Tampil nilai total harga berada dibawah <i>tab – tab</i> menu pilihan	Total harga pesanan telah muncul	<b>Berhasil</b>
6	Membatalkan pesanan	Membatalkan pesanan yang telah dipesan dan pesanan tersebut belum dimasak	Android 4.0.1	Halaman <i>order</i>	Klik nama makanan atau minuman yang ingin dibatikan	Klik batal	Terdapat tanda icon  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
7	Melihat pesanan yang sedang diproses	Melihat tanda makanan atau minuman yang telah diproses oleh pihak dapur	Android 4.0.1	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain kemudian klik <i>tab order</i> kembali	-	Terdapat tanda icon  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
8	Melihat pesanan yang telah selesai dimasak	Melihat tanda makanan atau minuman yang telah selesai dimasak	Android 4.0.1	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain kemudian klik <i>tab order</i> kembali	-	Terdapat tanda icon  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
9	Melihat artikel	Melihat artikel – artikel mengenai restoran, gizi	Android 4.0.1	-	Klik <i>tab</i> artikel	-	Menampilkan <i>list</i> artikel yang terbaru	<i>List</i> artikel terbaru ditampilkan	<b>Berhasil</b>

		kesehatan dan lainnya							
10	Melihat Outlet	Melihat <i>outlet</i> lain atau cabang lain restoran	Android 4.0.1	-	Klik <i>tab outlet</i>	-	Menampilkan <i>list outlet</i>	Tampilan nama <i>outlet</i> serta gambar dan alamat lengkap	<b>Berhasil</b>

Table 5.2. Pengujian aplikasi pada *smartphone* Blackberry 9780 dengan metode *Blackbox*

NO	Nama Pengujian	Deskripsi	Device	Pre kondisi	Prosedur Pengujian	Masukan	Hasil yang diharapkan	Hasil yang didapat	Ket
1	Membuka aplikasi	Menjalankan Aplikasi yang telah dipasang di <i>smartphone</i>	Blackberry 9780	-	Klik <i>Icon Daily Menu</i>	-	Menampilkan halaman <i>login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	Tampilan <i>form login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	<b>Berhasil</b>
2	Melakukan login	Pengujian untuk masuk kehalaman menu makanan dan minuman dengan login terlebih dahulu	Blackberry 9780	Halaman <i>Login</i>	Mengisi <i>form login</i>	Masukkan <i>username, password</i> dan <i>passkey</i>	Masuk ke tampilan halaman <i>menu Makanan dan minuman</i>	Tampilan halaman menu makanan dan minuman	<b>Berhasil</b>
3	Memesan makanan atau	Melakukan pemesanan	Blackberry 9780	Halaman <i>menu</i>	Memilih makanan atau	Klik makanan	Makanan atau minuman yang	Makanan atau minuman yang	<b>Berhasil</b>

	minuman	makanan atau minuman		makanan dan minuman	minuman dengan klik salah satunya.	atau minuman yang akan di pilih	dipilih tampil pada <i>tab order</i>	dipilih berhasil	
4	Melihat Pemesanan	Melihat daftar menu yang telah dipesan	Blackberry 9780	Halaman Menu	Klik <i>tab order</i>	-	Menampilkan daftar <i>list</i> yang telah dipesan sebelumnya	Tampilan list yang telah dipesan	<b>Berhasil</b>
5	Melihat total harga pesanan	Nilai total harga yang telah dipesan	Blackberry 9780	-	Klik <i>tab order</i> atau klik <i>tab</i> lainnya	-	Tampil nilai total harga berada dibawah <i>tab – tab</i> menu pilihan	Total harga pesanan telah muncul	<b>Berhasil</b>
6	Membatalkan pesanan	Membatalkan pesanan yang telah dipesan dan pesanan tersebut belum dimasak	Blackberry 9780	Halaman <i>order</i>	Klik nama makanan atau minuman yang ingin dibatikan	Klik batal	Terdapat tanda <i>icon</i>  disebelah nama menu	<i>Icon</i> tidak berubah	<b>Tidak Berhasil</b>
7	Melihat pesanan yang sedang diproses	Melihat tanda makanan atau minuman yang telah diproses oleh pihak dapur	Blackberry 9780	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain kemudian klik <i>tab order</i> kembali	-	Terdapat tanda <i>icon</i>  disebelah nama menu	<i>Icon</i> tidak berubah	<b>Tidak Berhasil</b>
8	Melihat pesanan yang telah selesai	Melihat tanda makanan atau	Blackberry 9780	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain	-	Terdapat tanda <i>icon</i>  disebelah	<i>Icon</i> tidak berubah	<b>Tidak Berhasil</b>



	dimasak	minuman yang telah selesai dimasak			kemudian klik <i>tab order</i> kembali		nama menu		
9	Melihat artikel	Melihat artikel – artikel mengenai restoran, gizi kesehatan dan lainnya	Blackberry 9780	-	Klik <i>tab</i> artikel	-	Menampilkan <i>list</i> artikel yang terbaru	<i>List</i> artikel terbaru ditampilkan	<b>Berhasil</b>
10	Melihat Outlet	Melihat <i>outlet</i> lain atau cabang lain restoran	Blackberry 9780	-	Klik <i>tab outlet</i>	-	Menampilkan <i>list outlet</i>	Tampilan nama <i>outlet</i> serta gambar dan alamat lengkap	<b>Berhasil</b>

Tabel 5.1 dan tabel 5.2 berjalan dengan *native* aplikasi *mobile-resto reservation*. Aplikasi juga dapat berjalan pada *web browser* bawaan dengan cara mengetik alamat aplikasi tersebut. Berikut tabel pengujian jika dijalankan pada *web browser* masing - masing *smartphone*.

Table 5.3. Pengujian *via web* pada *smartphone* Android 4.0.1 dengan metode *Blackbox*

NO	Nama Pengujian	Deskripsi	Device	Pre kondisi	Prosedur Pengujian	Masukan	Hasil yang diharapkan	Hasil yang didapat	Ket
1	Membuka aplikasi	Menjalankan	Web	-	Ketik alamat	-	Menampilkan	Tampilan <i>form</i>	<b>Berhasil</b>

		Aplikasi yang telah dipasang di <i>smartphone</i>	Android 4.0.1		URL aplikasi		halaman <i>login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	<i>login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	
2	Melakukan <i>login</i>	Pengujian untuk masuk kehalaman menu makanan dan minuman dengan <i>login</i> terlebih dahulu	Web Android 4.0.1	Halaman <i>Login</i>	Mengisi <i>form login</i>	Masukkan <i>username, password</i> dan <i>passkey</i>	Masuk ke tampilan halaman <i>menu Makanan</i> dan <i>minuman</i>	Tampilan halaman menu makanan dan minuman	<b>Berhasil</b>
3	Memesan makanan atau minuman	Melakukan pemesanan makanan atau minuman	Web Android 4.0.1	Halaman <i>menu makanan</i> dan <i>minuman</i>	Memilih makanan atau minuman dengan klik salah satunya.	Klik makanan atau minuman yang akan di pilih	Makanan atau minuman yang dipilih tampil pada <i>tab order</i>	Makanan atau minuman yang dipilih berhasil	<b>Berhasil</b>
4	Melihat Pemesanan	Melihat daftar menu yang telah dipesan	Web Android 4.0.1	Halaman <i>Menu</i>	Klik <i>tab order</i>	-	Menampilkan daftar <i>list</i> yang telah dipesan sebelumnya	Tampilan <i>list</i> yang telah dipesan	<b>Berhasil</b>
5	Melihat total harga pesanan	Nilai total harga yang telah dipesan	Web Android 4.0.1	-	Klik <i>tab order</i> atau klik <i>tab</i> lainnya	-	Tampil nilai total harga berada dibawah <i>tab – tab</i> menu pilihan	Total harga pesanan telah muncul	<b>Berhasil</b>







6	Membatalkan pesanan	Membatalkan pesanan yang telah dipesan dan pesanan tersebut belum dimasak	Web Android 4.0.1	Halaman <i>order</i>	Klik nama makanan atau minuman yang ingin dibatalkan	Klik batal	Terdapat tanda icon  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
7	Melihat pesanan yang sedang diproses	Melihat tanda makanan atau minuman yang telah diproses oleh pihak dapur	Web Android 4.0.1	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain kemudian klik <i>tab order</i> kembali	-	Terdapat tanda icon  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
8	Melihat pesanan yang telah selesai dimasak	Melihat tanda makanan atau minuman yang telah selesai dimasak	Web Android 4.0.1	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain kemudian klik <i>tab order</i> kembali	-	Terdapat tanda icon  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
9	Melihat artikel	Melihat artikel – artikel mengenai restoran, gizi kesehatan dan lainnya	Web Android 4.0.1	-	Klik <i>tab</i> artikel	-	Menampilkan <i>list</i> artikel yang terbaru	<i>List</i> artikel terbaru ditampilkan	<b>Berhasil</b>
10	Melihat Outlet	Melihat <i>outlet</i> lain atau cabang lain restoran	Web Android 4.0.1	-	Klik <i>tab outlet</i>	-	Menampilkan <i>list outlet</i>	Tampilan nama <i>outlet</i> serta gambar dan alamat lengkap	<b>Berhasil</b>

Table 5.4. Pengujian *via web* pada *smartphone* Blackberry 9780 dengan metode *Blackbox*

NO	Nama Pengujian	Deskripsi	Device	Pre kondisi	Prosedur Pengujian	Masukan	Hasil yang diharapkan	Hasil yang didapat	Ket
1	Membuka aplikasi	Menjalankan Aplikasi yang telah dipasang di <i>smartphone</i>	Web Blackberry 9780	-	Ketik URL alamat aplikasi	-	Menampilkan halaman <i>login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	Tampilan <i>form login</i> berisi <i>text view username, password</i> dan <i>passkey</i>	<b>Berhasil</b>
2	Melakukan login	Pengujian untuk masuk kehalaman menu makanan dan minuman dengan login terlebih dahulu	Web Blackberry 9780	Halaman <i>Login</i>	Mengisi <i>form login</i>	Masukkan <i>username, password</i> dan <i>passkey</i>	Masuk ke tampilan halaman <i>menu Makanan dan minuman</i>	Tampilan halaman menu makanan dan minuman	<b>Berhasil</b>
3	Memesan makanan atau minuman	Melakukan pemesanan makanan atau minuman	Web Blackberry 9780	Halaman <i>menu makanan dan minuman</i>	Memilih makanan atau minuman dengan klik salah satunya.	Klik makanan atau minuman yang akan di pilih	Makanan atau minuman yang dipilih tampil pada <i>tab order</i>	Makanan atau minuman yang dipilih berhasil	<b>Berhasil</b>
4	Melihat Pemesanan	Melihat daftar menu yang telah	Web Blackberry	Halaman <i>Menu</i>	Klik <i>tab order</i>	-	Menampilkan daftar <i>list</i> yang telah	Tampilan list yang telah dipesan	<b>Berhasil</b>

		dipesan	9780				dipesan sebelumnya		
5	Melihat total harga pesanan	Nilai total harga yang telah dipesan	Web Blackberry 9780	-	Klik <i>tab order</i> atau klik <i>tab</i> lainnya	-	Tampil nilai total harga berada dibawah <i>tab – tab</i> menu pilihan	Total harga pesanan telah muncul	<b>Berhasil</b>
6	Membatalkan pesanan	Membatalkan pesanan yang telah dipesan dan pesanan tersebut belum dimasak	Web Blackberry 9780	Halaman <i>order</i>	Klik nama makanan atau minuman yang ingin dibatikan	Klik batal	Terdapat tanda <i>icon</i>  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
7	Melihat pesanan yang sedang diproses	Melihat tanda makanan atau minuman yang telah diproses oleh pihak dapur	Web Blackberry 9780	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain kemudian klik <i>tab order</i> kembali	-	Terdapat tanda <i>icon</i>  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
8	Melihat pesanan yang telah selesai dimasak	Melihat tanda makanan atau minuman yang telah selesai dimasak	Web Blackberry 9780	Halaman <i>order</i>	<i>Refresh</i> atau klik <i>tab</i> lain kemudian klik <i>tab order</i> kembali	-	Terdapat tanda <i>icon</i>  disebelah nama menu	<i>Icon</i> telah berubah	<b>Berhasil</b>
9	Melihat artikel	Melihat artikel – artikel mengenai restoran, gizi	Web Blackberry 9780	-	Klik <i>tab</i> artikel	-	Menampilkan <i>list</i> artikel yang terbaru	<i>List</i> artikel terbaru ditampilkan	<b>Berhasil</b>

		kesehatan dan lainnya							
10	Melihat Outlet	Melihat <i>outlet</i> lain atau cabang lain restoran	Web Blackberry 9780	-	Klik <i>tab outlet</i>	-	Menampilkan <i>list outlet</i>	Tampilan nama <i>outlet</i> serta gambar dan alamat lengkap	<b>Berhasil</b>

Untuk melihat tabel pengujian dari akun lain dapat dilihat pada lampiran D.

Keterangan *Icon* :



= Dalam Proses Tunggu



= Sedang Dimasak



= Selesai Dimasak



= Pesanan Dibatalkan

## BAB VI

### PENUTUP

#### 6.1. Kesimpulan

Setelah menyelesaikan serangkaian tahapan dalam merancang dan membangun aplikasi *mobile-resto reservation* berbasis *client-server* pada *smartphone* Android dan Blackberry yang dimulai dari pengumpulan data, teknologi Android dan Blackberry hingga pada tahapan pengujian, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Aplikasi *mobile-resto reservation* dapat berjalan pada perangkat *smartphone* Android dan Blackberry sesuai perancangan.
2. Aplikasi *client* dapat melakukan pemesanan hingga melihat pesanan sudah diproses atau belum hingga selesai.
3. Terdapat beberapa fitur yang tidak bekerja pada *platform* Blackberry.
4. Aplikasi *mobile-resto reservation* membantu proses bisnis penjualan restoran dengan adanya fitur – fitur yang menunjang dalam bisnis.
5. Menerapkan teknologi baru dalam pemesanan akan membuat citra khusus bagi pelanggan.
6. Teknologi baru dalam pemesanan makanan akan membuat bisnis restoran menjadi maju dan diminati pelanggan, khususnya bagi pelanggan yang *electronic life* (e-life).

## 6.2. Saran

Beberapa hal yang disarankan dalam pengembangan aplikasi aplikasi *mobile-resto reservation* berbasis *client server* pada *smartphone* Android dan Blackberry ini adalah sebagai berikut:

1. Pada pengembangan aplikasi *mobile-resto reservation* selanjutnya diharapkan sudah mampu berjalan pada *device* lain selain *smartphone* Android dan Blackberry.
2. Pada pengembangan aplikasi *mobile-resto reservation* selanjutnya diharapkan tidak hanya mampu memesan makanan dengan pilihan inputan, juga mampu dengan pemesanan dengan suara dengan teknologi *speech recognition*.
3. Pada pengembangan aplikasi *mobile-resto reservation* selanjutnya diharapkan memiliki lebih banyak fitur lain, seperti adanya *game* untuk menghabiskan waktu pelanggan dalam menunggu pesanan selesai dimasak.
4. Bagi restoran yang menilai sebuah kepuasan pelanggan dapat membuat fitur *member* pada aplikasi *mobile-resto reservation* tanpa *passkey* untuk memesan makanan dan minuman selanjutnya.